N72-17006

NASA CONTRACTOR REPORT

NASA CR-1979

NASA CR-1979

A STUDY OF THE MARKOV GAME APPROACH TO TACTICAL MANEUVERING PROBLEMS

by Sheldon Baron, David L. Kleinman, and Saul Serben

Prepared by
BOLT BERANEK AND NEWMAN, INC.
Cambridge, Mass. 02138
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • FEBRUARY 1972

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-1979		

4. Title and Subtitle	5. Report Date
A Study of the Markov Game Approach to Tactical Maneuvering Problems	February 1972
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Sheldon Baron, David L. Kleinman, and Saul Serbin	2179
	10. Work Unit No.
	136-62-01-02
9. Performing Organization Name and Address	11. Contract or Grant No.
Bolt Beranek and Newman, Inc. 50 Moulton Street Cambridge, MA 02138	NAS1-9910
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The results of a study to apply a Markov game approach to planar air combat problems are presented in this report. The underlying approach is reviewed and a sophisticated computer program (MAGPIE) developed to solve the planar problems is discussed. Numerical results for highly idealized versions of the problem are presented with a view towards improving understanding of the basic approach. Typical results in the form of optimal costs, strategies and trajectories are also obtained for a realistic version of the planar combat problem. The solution to this problem demonstrates the feasibility of using the Markov game approach for solving meaningful problems. Analysis indicates, however, that straightforward extension to three-dimensional air combat problems may be impractical from the standpoint of the computation time required. Alternative approaches are therefore suggested.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Differential games and aerial combat	Unclassified-unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	83	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

FOREWORD

This report was prepared by the Control Systems Department of Bolt Beranek and Newman Inc., Cambridge, Massachusetts. It represents the results of a study made for the Langley Research Center under NASA Contract No. NAS1-9910 and is a continuation of the work done under NASA Contract No. NAS1-8296. The work was administered under the direction of Dr. John D. Bird of the Theoretical Mechanics Branch, FDCD at Langley.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

LIST OF ILLUSTRATIONS

SYMBOLS

$\underline{x}$	state-vector
S_i	discrete state-block
p_{ij}	transition probability
c_{ij}	incremental cost
$V(i)$	total cost for S_i
π_u	attacker strategy
π_v	evader strategy
r	range, ft.
ξ	azimuth, rad
ψ	relative heading angle, radians
V_a	absolute velocity of attacker, ft/sec
V_e	absolute velocity of evader, ft/sec
M	Mach No. $(=V/a_s)$
n	normal acceleration $(=q\,\frac{SC_L}{W})$, g's
g	gravitational acceleration constant, 32.2 ft/sec^2
a_s	speed of sound, ft/sec
T_a	thrust of attacker, lbs
T_e	thrust of evader, lbs
C_L	nondimensional lift coefficient
S	wing area, ft^2
W	airplane weight, lbs
q	dynamic pressure, $1/2\rho V^2$, lbs/ft^2
C_D	nondimensional drag coefficient
C_{D_o}	drag coefficient at zero angle of attack

$C_{D_{C_L^2}}$	slope of C_D vs. C_L^2
n_{max}	maximum normal acceleration
$C_{L_{max}}$	maximum C_L
T_{max}	maximum thrust
u_1	attacker turning control
u_2	attacker thrust control
v_1	evader turning control
v_2	evader thrust control
$( )_a$	subscript a refers to attacker
$( )_e$	subscript e refers to evader

A STUDY OF THE MARKOV GAME APPROACH TO TACTICAL MANEUVERING PROBLEMS

By Sheldon Baron, David L. Kleinman and Saul Serben
Bolt Beranek and Newman Inc.
Cambridge, Massachusetts 02138

1. INTRODUCTION

Problems concerning the tactical maneuvering of aircraft have assumed increasing importance in recent years. Insofar as these problems involve dynamic systems in competition, it is natural to attempt to apply "differential game theory" to their solution. However, realistic aerial combat problems, involving nonlinear dynamics and imperfect information, appear to be beyond solution by state-of-the-art differential game methods.

In a recent study (Ref.1), we suggested a novel conceptualization of the pursuit-evasion, manned aerial combat problem along with a corresponding computational scheme for its solution. In essence, our approach was to formulate the problem as a Markov game by discretizing the state and control spaces in a manner dictated by physical considerations. The Markov game problem could then be solved iteratively. The method accommodated nonlinear dynamics easily and the lack of perfect information was accounted for directly in the problem formulation. Furthermore, the computation scheme yielded optimal <u>feedback</u> strategies for the problem.

The Markov game approach to aerial combat problems seemed to have considerable potential but it had not been adequately tested. In particular, there was a serious question as to the practicality of the method because it imposed large computational demands. This study was undertaken to obtain a better appraisal of the Markov approach. Our prime objective was to apply the technique to a reasonably complex aerial combat problem so that we could assess the computational problems more realistically. Secondary goals were to increase basic understanding of the techniques and to develop improved computational procedures.

To fulfill our aims, we developed a sophisticated, relatively general computer program for solving Markov games of aerial combat. The program, called MAGPIE, was first used to obtain solutions to the discrete versions of the Homocidal Chauffeur and Two-Car problems solved previously in Reference 1 with a special purpose program; this provided a basic check of MAGPIE. Numerical investigations of these highly idealized planar combat problems allowed us to explore analytic and computational aspects of the Markov approach in a reasonably well-understood, cost-effective context.

We then used the MAGPIE program to solve a variable-speed, planar combat problem. This was a five state-variable problem with equations of motion that were highly nonlinear. To our knowledge, this is the most complex dynamic game for which a "solution" has been obtained. However, extrapolating these results leads us to conclude that a realistic, three-dimensional problem is beyond current capability. We therefore suggest some modifications to the basic approach that are likely to result in useful answers to meaningful problems of a somewhat reduced scope.

The report is organized as follows. Chapter 2 contains a brief review of the Markov game approach to manned aerial combat problems. An overview of the MAGPIE program is presented in Chapter 3. The numerical investigations of the Homocidal Chauffeur and Two-Car problems are discussed in Chapter 4. The Markov game approach is applied to the variable speed planar combat problem in Chapter 5. In Chapter 6, we examine the computational feasibility of the Markov approach, in light of the results of applying MAGPIE to planar combat problems. Concluding remarks and suggestions for further work are contained in Chapter 7.

2. BACKGROUND

In this chapter we summarize the Markov game approach to manned aerial combat problems as developed in Reference 1. We outline the formulation of the problem as a Markov game and the computational solution to this game. We also discuss briefly some problem areas, raised in Reference 1, that gave impetus and direction to this work. We emphasize motivation and underlying concepts here; mathematical details may be found in Reference 1.

Problem Formulation

The "computational" scheme that we have developed for studying manned aerial combat problems is based primarily on two physical concepts. The first relates to the information upon which the pilots must make their control decisions. The second relates to the nature of the decisions themselves.

An engagement between two vehicles can be described mathematically in some state-space $\underline{X}$ by the time evolution of a state trajectory $\underline{x}(t)$, as governed by appropriate dynamical equations of motion. In general, the components of $\underline{x}(t)$ would represent the aircrafts' positions, velocities, etc. at some time t. Conceptually, but rarely in practice, the knowledge of $\underline{x}(t)$ coupled with the vehicles' equations of motion and the objective of the encounter, is all that is needed by "differential game theory" to provide optimal pursuit and evasion control strategies. Unfortunately, in manned aerial combat situations the pilots rarely have precise knowledge of the state $\underline{x}(t)$ and must base their control actions on imprecise estimates of the current state of play.

We attempt to include this physical constraint directly into the problem formulation. Our approach is to decompose the state space $\underline{X}$ into disjoint blocks S_0, $S_1, \ldots, S_N$. These blocks may be of arbitrary size and shape and satisfy

$$\underline{X} = \bigcup_{i=0}^{N} S_i$$

We assume that both players know the system state only to within a block S_i. Thus, a player cannot discern where the state is within S_i, but only the fact that $\underline{x}(t)$ is somewhere (with, e.g., uniform probability) in the known block. As time evolves, the state trajectory will make transitions from block to block. It is assumed that only these transitions are perceived by the players.* The blocks S_i may be thought of as "perceived states" for a discretized game. In general, the state-space decomposition should reflect physical considerations inherent in the problem (e.g., intrinsic problem geometry) and human limitations relating to perception and/or estimation and information processing.

The state transitions from block to block are influenced by the particular control actions taken by the players. Often, these control actions can also be discretized in a meaningful way. In our approach, we assume the control strategies of the players are constructed from a finite set of "canonical control maneuvers." The "maneuvers" in this set could be basic actions, such as various g-turns, -pull-ups and -dives. The set could also include more specialized or complex maneuvers such as a "scissors" or a "yo-yo."

*The assumption that a "transition" is perceived "perfectly" is consistent with the notion of a uniform probability distribution within a block. A different set of assumptions is possible and may prove to be desirable in some instances.

In any case, we introduce into the problem formulation a finite set of control choices for the pursuer and evader, denoted respectively by $U_\alpha = \{\underline{u}^1, \underline{u}^2, \ldots, \underline{u}^\alpha\}$ and $V_\beta = \{\underline{v}^1, \ldots, \underline{v}^\beta\}$.

Suppose that a particular pair of maneuvers $(\underline{u}\epsilon U_\alpha, \underline{v}\epsilon V_\beta)$ have been selected by the players. Then, a given state $\underline{x}(t)$ would evolve to some new value under the influence of the dynamic equations of motion. However, the "actual" state $\underline{x}(t)$ is not known to the players; all that is known is that $\underline{x}(t)$ is in some block, say S_i. Hence, a particular trajectory cannot be "followed" and the most that one can hope to determine are the probabilities of transitions between blocks, under the action of a particular maneuver pair. Therefore, we define

$$p_{ij}(\underline{u},\underline{v}) \triangleq \begin{array}{l}\text{probability of a transition from block } S_i \\ \text{to block } S_j \text{ with maneuvers } (\underline{u},\underline{v})\end{array}$$

Thus, the nature of our discretizing assumptions reduces the dynamic situation to one of a controllable Markov chain.

To formulate a Markov game, we must stipulate the goals of the players. In pursuit-evasion problems, the attacker's goal may be expressed in terms of placing the target in some highly desirable (capture) state at minimum "cost". On the other hand, it is assumed that the evader wishes to avoid capture if possible, or to maximize the cost of capture.* Thus, to quantify the players' goals, we must define the capture state and the cost. For convenience, we let the capture state correspond to the first state-block S_o. Then, by definition,

$$p_{oi} = \begin{cases} 0 & i \neq 0 \\ 1 & i = 0 \end{cases}$$

*This is, of course, the zero-sum game situation.

We associate an incremental cost with each transition between blocks, defining

$c_{ij}(\underline{u},\underline{v}) \triangleq$ the *average* cost of a transition from S_i to S_j using the maneuver pair $(\underline{u},\underline{v})$.

Notice that we must use average or expected costs because of the uncertainty in the state.

The total expected cost will depend on the initial state and on the strategies employed by the players. A u-strategy, π_u, is defined as a set of N allowable control maneuvers, $\underline{u}(i)\epsilon U_\alpha(i=1,\ldots,N)$, such that the maneuver $\underline{u}(i)$ is applied whenever the state of the game is S_i. A similar definition applies to the evader's strategy, π_v. Thus, strategies are simply feedback laws defined over the discrete space: $\bigcup_{i=1}^{N} S_i$. Denoting by $V^\pi(i)$ the total cost incurred for a game starting in S_i and played with policy pair $\pi = (\pi_u,\pi_v)$, we have

$$V^\pi(i) = \sum_{i=1}^{N} p_{ij}(\underline{u}(i),\underline{v}(i))[V^\pi(j) + c_{ij}(\underline{u}(i),\underline{v}(i))] \qquad (1)$$

The optimal strategy pair, in a game-theoretic sense, is the pair $\pi^* = (\pi_u^*,\pi_v^*)$ for which

$$V^{\pi^*}(i) = \underset{\pi_u}{\text{Min}}\ \underset{\pi_v}{\text{Max}}\ V^\pi(i) = \underset{\pi_v}{\text{Max}}\ \underset{\pi_u}{\text{Min}}\ V^\pi(i), \text{ for all } i \qquad (2)$$

Given the nature of the state and control space discretizations, we see how the solution to the Markov game might reflect a battle situation in which the pilots select particular "stylized" maneuvers (perhaps learned in training), that best serve their respective goals, based on a relatively crude assessment of their situation.

Method of Solution

There are two basic elements in the solution of aerial-combat Markov game described above. First, the transition probabilities $p_{ij}(\underline{u},\underline{v})$ must be determined. Second, given the transition probability matrix, the Markov game must be solved to find the optimal strategies.

Both aspects of the problem solution are greatly benefitted by assuming that once the players choose their control maneuvers $\underline{u}$, $\underline{v}$ in a given block S_i, they must continue to employ these maneuvers until the state is perceived to be in a new block. Once a block transition is perceived the players may change to a new maneuver. This assumption is appealing from a physical viewpoint since control decisions are thus made on the basis of perceived information, which, in this case, changes only when block transitions occur. In other words, a new decision is made based on the perceived outcome of the old decision. From a mathematical viewpoint, the assumption implies that, in one stage, state transitions are restricted to only adjacent blocks before the players reoptimize over their sets of allowable maneuvers. In consequence, the storage requirements and computational demands are substantially less than those of a conventional Markov game formulation in which transitions between *any* two states are allowed.

Only in rare cases will it be possible to determine the transition probabilities analytically. However, they may be computed approximately using a Monte-Carlo type method. M points are distributed uniformly over a given block S_i and, for a given maneuver pair ($\underline{u},\underline{v}$), the equations of motion are integrated forward in time, starting from each of the M points. The integration is continued until the state trajectory enters an adjacent block.* If M_j is the number of points initially within S_i that are driven to S_j (with the given maneuvers $\underline{u}$ and $\underline{v}$), then $p_{ij}(\underline{u},\underline{v}) \approx M_j/M$. In the course of this computation one can also determine the average time of a transition from S_i to S_j, a quantity that is often needed for computing "costs".

Once the transition probabilities have been calculated, there remains the problem of finding the π^* that satisfies Eq.(2). A relatively new numerical technique (Refs.2,3), based on the Gauss-Seidel method for solving linear equations, provides a convergent algorithm for determining the optimal-strategies and -expected cost** The salient feature of the computational scheme is a monotone convergence that is guaranteed to be more rapid than straightforward dynamic programming.

Problem Areas

Reference 1 demonstrated the potential of the Markov approach to aerial combat problems, but it left unresolved some important problems. The key problem concerns computational requirements. The necessity of computing the transition probabilities $p_{ij}(\underline{u},\underline{v})$ for all i, $\underline{u}$ and $\underline{v}$ imposes the largest burden. While each computation may be relatively simple in itself, a realistic problem

*Or until a predesignated time has elapsed.

**When the game has no saddle-point solution, the algorithm will yield min-max (or, if desired, max-min) solutions.

could involve an enormous number of such calculations. A related problem is the storage of the p_{ij}'s. It is unlikely that they could be stored in core and reading them from other storage devices increases the effective computation time. The iterative solution of the discrete Markov game presents a lesser, but nontrivial, computational demand. In the examples solved in Reference 1 approximately one-third of the total computation time was devoted to the iterative procedure.

The magnitude of the computational requirements raises a question as to how realistic a problem it is feasible to solve by this method. A corollary issue concerns methods for reducing the computational load. The results of Reference 1 did not shed much light on these problems. The examples solved there, namely the familiar Homocidal Chauffeur and Two-Car games involved comparatively simple dynamic situations. The computer programs used in their solution were quite specialized and there was no opportunity to explore means for improving or optimizing the techniques. Thus, it was not possible to make reliable extrapolations to realistic problems.

A number of issues of a more analytic nature require further study. The effects of changes in state-space discretization are of considerable interest. In particular, one would like to know if the solution to the Markov game approaches that of a corresponding differential game as the discretization becomes finer. Another, perhaps related, question concerns the existence of a saddle-point solution to the discrete Markov game. Even if the continuous game has a pure strategy saddle-point, we cannot be certain that it will be retained in the discretized game. Finally, vehicle trajectories based on "playing out" the optimal strategies would significantly enhance our understanding of the approach; such trajectories were not obtained in Reference 1.

3. MAGPIE

To solve a class of aerial combat problems, we developed a fairly general, essentially machine-independent, computer program called MAGPIE (for MArkov Game Planar Intercept-Evasion Package). The development of this sophisticated program consumed the major part of our total effort in this study. Here, we present an overview of the program, highlighting some of the more interesting features it incorporates.

Program Structure

The structure of MAGPIE is illustrated in the flow chart of Figure 1. As can be seen, the program is modular in design. Actually, it consists of three major sub-programs, PROBABILITIES, MARKOV and TRAJECTORIES. Each of the sub-programs uses the same input program (INPUT). They can be run independently provided they are run sequentially; in particular, MARKOV requires the output of PROBABILITIES and TRAJECTORIES requires the output of MARKOV.

The program includes features in PROBABILITIES and MARKOV, that allow the computation to be stopped, with pertinent results saved so that a re-start is possible. These "pick-up" features protect against the loss of large amounts of productive computation due to computer crashes. They also allow the problem to be run in segmented fashion, which is very important given the long computation times involved.

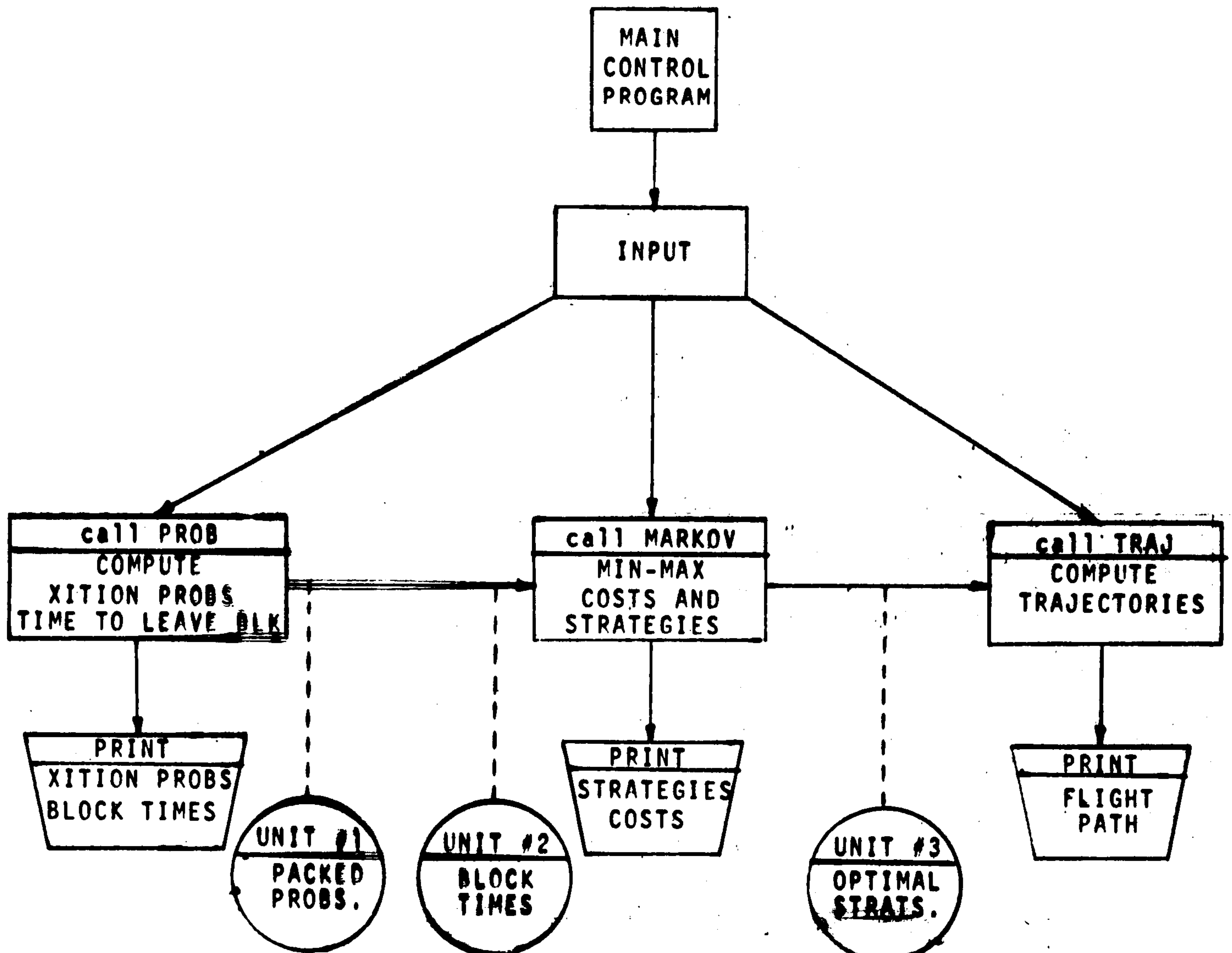

FIG. 1 OVERALL FLOW FOR MAGPIE

Input

There are five classes of inputs that comprise the INPUT program of MAGPIE: (1) alphanumeric inputs (title cards); (2) initialization parameters; (3) aerodynamic and thrust data tables; (4) state-space geometry parameters; and, (5) strategy tables. The form of these inputs was tailored somewhat to the dynamics for the Variable-Speed Planar Combat problem described in Chapter 5. However, the program modifications necessary to change to different dynamics (of higher order, say) are relatively minor.

From the standpoint of the problem, the most interesting aspect of the INPUT program relates to the state-space geometry parameters. The program was written for five-state variables. The number of blocks for each state variable was an input parameter, with a maximum of 20 allowed for each state. The state discretization could be nonlinear, as defined by the equation

$$(S_jB)_{i+1} = (S_jB)_i + L_j{*}K_j^{i-1}; \; i=1,\ldots,N_j \tag{3}$$

where $(S_jB)_i$ is the location of the inner boundary of the i-th block of the j-th state-variable; L_j is the length of the first block; K_j is a "stretch factor"; and N_j is the number of blocks in the discretization of the j-th state-variable. $(S_jB)_1$, L_j, K_j and N_j were input parameters; after they were chosen, the program computed the appropriate state-space discretization and produced a table of that discretization on output.

As part of the state-space geometry parameters, one specifies the number of points per state-block to be used in computing the transition probabilities (i.e., M in Chapter 2). The program then calculates appropriate point-locations as follows: the points are spread uniformly over the range of the state-variable and then the point locations are "stretched" in accordance with the above transformation (Eq.3).

"PROBABILITIES"

This sub-program computes the transition probabilities and the average time to leave a block. To accomplish this the equations of motion for the problem must be integrated. A sub-routine EULER performs this integration, using a first-order (Euler) scheme. This simple integrator should suffice because of the short integration times involved in a block transition. However, to improve accuracy and to save time where possible, a "regularized" time step is employed.* Thus, the time step for a given state-block was chosen as

$$\Delta t = \underset{i}{\mathrm{Min}} \frac{(\Delta x_i)}{2T(\dot{x}_i)} \tag{4}$$

where (Δx_i) is the corresponding block length for the i-th state-variable, $\dot{x}_i$ is the rate-of-change of that variable evaluated at the center of the block and T is an input parameter that is an indication of the "average" number of steps to leave the block.

Another interesting feature of the PROBABILITIES program involves the treatment of "corner-transitions." By a "corner-transition," we mean a transition between two n-dimensional state-blocks that are "adjacent" in the sense that they have a common (n-2)- or lower-dimensional edge. For example, in the two-dimensional situation illustrated in Figure 2, $S_5 \to S_1, S_3, S_7$ or S_9 are corner-transitions.

*A fixed time-step can be selected, at the user's option.

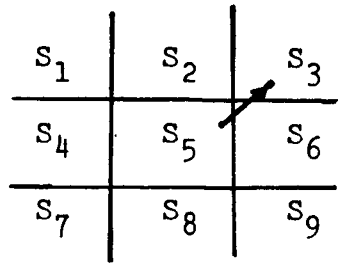

FIG. 2 A "CORNER" TRANSITION IN TWO-DIMENSIONS

The probability of a true corner-transition is virtually zero.* However, because of the discrete integration,a situation that would be interpreted as a corner-transition can occur (See Fig.2). The PROBABILITIES program corrects for this situation by "apportioning" among allowable transitions. Thus, in the case shown in Fig. 2, it would be assumed that two points were integrated and that transitions to S_2 *and* to S_6 had occurred. In addition, the program would adjust the average time for a transition and it would keep a tally of the number of points that appeared to undergo a corner-transition. This latter number is, in some sense, a check on the adequacy of the integration time-step.

Finally, it should be mentioned that the computed probabilities are "packed" for auxiliary storage. The number of probabilities per "word" is an input option — an important feature for machine independence. For the CDC-6600,we packed eight probabilities to a word. The word-packing scheme saves storage space and CPU-time in reading the stored probabilities.

---

*This was first called to our attention by Dr. John Bird of NASA-Langley Research Center. Notice that eliminating corner transitions saves both storage and computation time.

"MARKOV"

The program MARKOV takes the p_{ij}'s and c_{ij}'s obtained in PROBABILITIES and computes optimal costs and optimal strategies. Before the computation can proceed, capture conditions must be input to MARKOV. In the current program, capture is specified by three numbers; the first two numbers designate appropriate range (r) and azimuth (ξ) block numbers, defining a region in relative-position space that is highly-favorable to the attacker; the third number designates the time that the evader must remain in this region for "capture" to occur. Several points are worth noting in connection with the capture conditions. First, any block in r-ξ space may be chosen as a capture block. Indeed, several, non-contiguous blocks may be so designated. Secondly, although the current implementation does not involve an explicit constraint on relative heading or speed of the aircraft, such a constraint is implicit in the time-in-block specification. Moreover, it would be relatively simple to modify the program to define capture blocks in terms of more state-variables. Finally, it is also possible to designate blocks that are highly favorable to the evader (i.e., have a high cost associated with them). Thus, we see there is considerable flexibility in this aspect of the problem definition.

The computation of the optimal costs and strategies is a accomplished by a modified version of the iterative scheme used in Reference 1. Specifically, the iterative procedure is defined by

$$\tilde{V}^n(i) = (1-\omega) \min_u \max_v \left[\sum_{j=1}^{i-1} p_{ij}(u,v)\tilde{V}^n(j) + \sum_{j=i}^{N} p_{ij}(u,v)\tilde{V}^{n-1}(j) \right.$$

$$\left. + \sum_{j=1}^{N} p_{ij}(u,v)c_{ij}(u,v) \right] + \omega\tilde{V}^{n-1}(i) \qquad (5)$$

Equation (5) is the min-max version of the "accelerated" Gauss-Seidel procedure proposed by Kushner and Kleinman (Ref.4) for use in Markov control problems. We have altered the scheme of Reference 4 slightly in that the acceleration parameter ω is reduced automatically if iteration errors increase. The initial value of ω is an input to the program.

Another feature, designed to save computation time, was added to the iterative procedure. According to Eq.(5) a min max operation is performed at every iteration; this can be quite wasteful in the latter stages of the computation when the costs $V(i)$ are changing very slowly. Thus, we have modified the procedure of Eq.(5), so that a min max operation is not performed at every iteration step, when the iteration error is below some prespecified (input) tolerance. The number of iterations between min max operations is chosen automatically, as a function of iteration error. This scheme of not optimizing at every step is a modification of Howard's "iteration in policy space" (Ref.5).

MARKOV included one other option deserving mention. It was possible to reverse the min and max operations of Eq.(5) by an input trigger. Thus, one could compute either min-max or max-min costs and strategies.

"TRAJECTORIES"

TRAJECTORIES allows us to "play" the game using the optimal strategies. A set of initial conditions are input to the program, along with the state-space geometry, equations of motion, capture conditions and optimal strategies. The program determines the state-block corresponding to the initial conditions, selects the proper strategies and begins integration of the equations of motion. The integrator in TRAJECTORIES is a fourth-order Runge-Kutta scheme, inasmuch as better accuracy than that used in the p_{ij} calculation is desirable here. After each integration step the state variables are checked to determine their location in the discrete state-space, so that the proper strategies may be selected. The integration continues until either capture occurs or a prespecified "final" time is reached. Thus, TRAJECTORIES permits us to generate deterministic paths that result from playing optimal strategies that were obtained from probabilistic considerations.

An interesting and potentially important feature of the TRAJECTORIES program is the inclusion of an option that allows overiding of the optimal strategy in any block(s). It is therefore possible to examine the effects of playing nonoptimally or to evaluate a given strategy against an "optimal" opponent.

4. NUMERICAL INVESTIGATIONS

In Chapter 2 we noted that, after the study of Reference 1, there were several unresolved analytic questions concerning the Markov approach. Because of the many heuristic aspects of this approach, it is very difficult to investigate these questions theoretically. Consequently, we decided to study some of them numerically by using MAGPIE. Our basic context was provided by the Homocidal Chauffeur and Two-Car problems solved in Reference 1. We present some of the results of our investigations in this chapter.

Homocidal Chauffeur and Two-Car Problems

For ease of reference, we describe briefly these two classical problems, first posed by Isaacs (Reference 6). Two vehicles move in a plane at constant speed. One vehicle, the attacker, has a greater speed and turn radius than that of the vehicle he is chasing. The game ends when, and if, the distance between the two vehicles becomes less than a given capture radius L. The attacker attempts to minimize the time of capture; the evader attempts to maximize it.

The kinematic equations for such an encounter may be written in a rotating, attacker-centered coordinate system

$$\dot{r} = V_e \cos(\xi+\psi) - V_a \cos \xi$$

$$\dot{\xi} = -u \frac{V_a}{R_a} - \frac{V_e}{r} \sin(\xi+\psi) + \frac{V_a}{r} \sin \xi$$

$$\dot{\psi} = v \frac{V_e}{R_e} - u \frac{V_a}{R_a}$$

where: r and ξ are the range and azimuth of the evading vehicle; ψ is the angle between the velocity vectors of the two vehicles; $V_a(V_e)$ and $R_a(R_e)$ are the speed and turn radius, respectively, of the attacker (evader). The pursuer and evader each selects his rate-of-turn, by choosing $|u|\leq 1$ and $|v|\leq 1$. This is the game of Two-Cars. As R_e becomes smaller and smaller, the evader can change directions faster and faster. In the limit when $R_e=0$, the evader can change directions instantaneously or, what is the same, can select ψ. This limiting case is called the Homocidal Chauffeur problem.

In Reference 1, discrete Markov game versions of the above problems were formulated and solved. For the Homocidal Chauffeur Problem the playing-space was defined as

$$\underline{X} = \{r,\xi:\ r\leq 10,\ 0\leq\xi\leq 180°\}$$

A unit radial discretization (Δr) was chosen along with a 20° discretization of azimuth. The control choices for the attacker were $u=\pm 1,0$; the evader could choose $\psi=45°, 135°, 225°, 315°$. The parameter values used in Reference 1, and in this study, were: $L=1.0$, $V_a=1.0$, $V_e=.7$, $R_a=1.5$.

The discrete version of the Two-Car problem had a playing-space defined by

$$\underline{X} = \{r,\xi,\psi:\ r<10,\ 0\leq\xi\leq 360°,\ 0\leq\psi\leq 360°\}$$

Range was discretized as in the previous case but ξ was discretized according to clock angles. Thus, there were twelve ξ-blocks satisfying: $30j-15\leq\xi\leq 30j+15$, $j=1,\ldots,12$. The ψ-coordinate was discretized into four blocks defined by: $90j-45 \leq \psi \leq 90j+45$, $j=1,2,3,4$. It was felt that the angular discretizations reflected the kinds of judgements that human pilots might make. The control choices available to pursuer and evader were, respectively, $u=\pm 1,0$ and $v=\pm 1,0$. The parameter values selected were: $L=1$, $V_a=1.0$, $R_a=1.5$, $V_e=.8$, $R_e=.5$.

Effects of State-Space Discretization

The manner in which the state-space is discretized is a unique and crucial feature of our approach. Thus, it is important to understand the effects of changes in this discretization. The ease with which state-space geometry may be changed in MAGPIE greatly facilitated the investigation of these effects.

Of special interest from a theoretic standpoint is the relation of the discrete game to a corresponding differential game, as a function of discretization. For the Homocidal Chauffeur problem, perhaps the most significant aspect of this relation is concerned with the "so-called" barrier. This curve in r-ξ space is of fundamental importance in the solution of the differential game; it separates regions of different pursuit strategy and the min-max time-to-capture is discontinuous across it. In Reference 1, it was found that the corresponding discrete problem also had a strategy-barrier. However, this barrier was located "within" the continuous barrier, reflecting a more "conservative" pursuit strategy. To see how the strategy-barrier behaves with discretization, we used MAGPIE to compute solutions to the discrete version for the following:

a. Base Discretization*: $\Delta r = 1$; $\Delta\xi = 20°$

b. Half-Base Discretization: $\Delta r = 1/2$; $\Delta\xi = 10°$

c. Fine-Discretization: $\Delta r = 1/4$; $\Delta\xi = 5°$

d. Nonlinear Discretization: $1 \le r(2) \le 1.5 \le r(3) \le 2.5 \le r(4) \le 4.5 \le r(5) \le 8.5$;

$30j-15 \le \xi \le 30j+15$, $j=0,1,\ldots,6$

Figure 3 shows the results in terms of the strategy-barrier for cases a, b and d. The result for the fine-discretization is shown in Fig. 4. The irregularities in the barrier for the fine-discretization are probably a result of the fact that the iterative "solution" had not fully converged in this case. It is clear from Figs. 3-4 that the discrete barrier approaches the continuous barrier as the discretization becomes finer.

In Fig. 5 the optimal costs for $2 \le r \le 3$ are plotted as a function of azimuth angle for cases a-c above. The costs change most rapidly in the $20° \le \xi \le 40°$ region; that is precisely the area that the barrier passes through (See Fig.4). Thus, the Markov game solution is approximating the cost-discontinuity of the differential game and, as might be expected, the approximation gets better as the discretization becomes finer.

Changes in discretization level reflect changes in information quality in the Markov formulation. In the Homocidal Chauffeur, the results of Fig. 5 indicate that, for the region considered, the expected time-to-capture (i.e., the cost) increases as the discretization becomes coarser. This cost increase was found to hold for the entire state-space. Hence, in the Homocidal Chauffeur,

---

*That is, the discretization of Ref. 1.

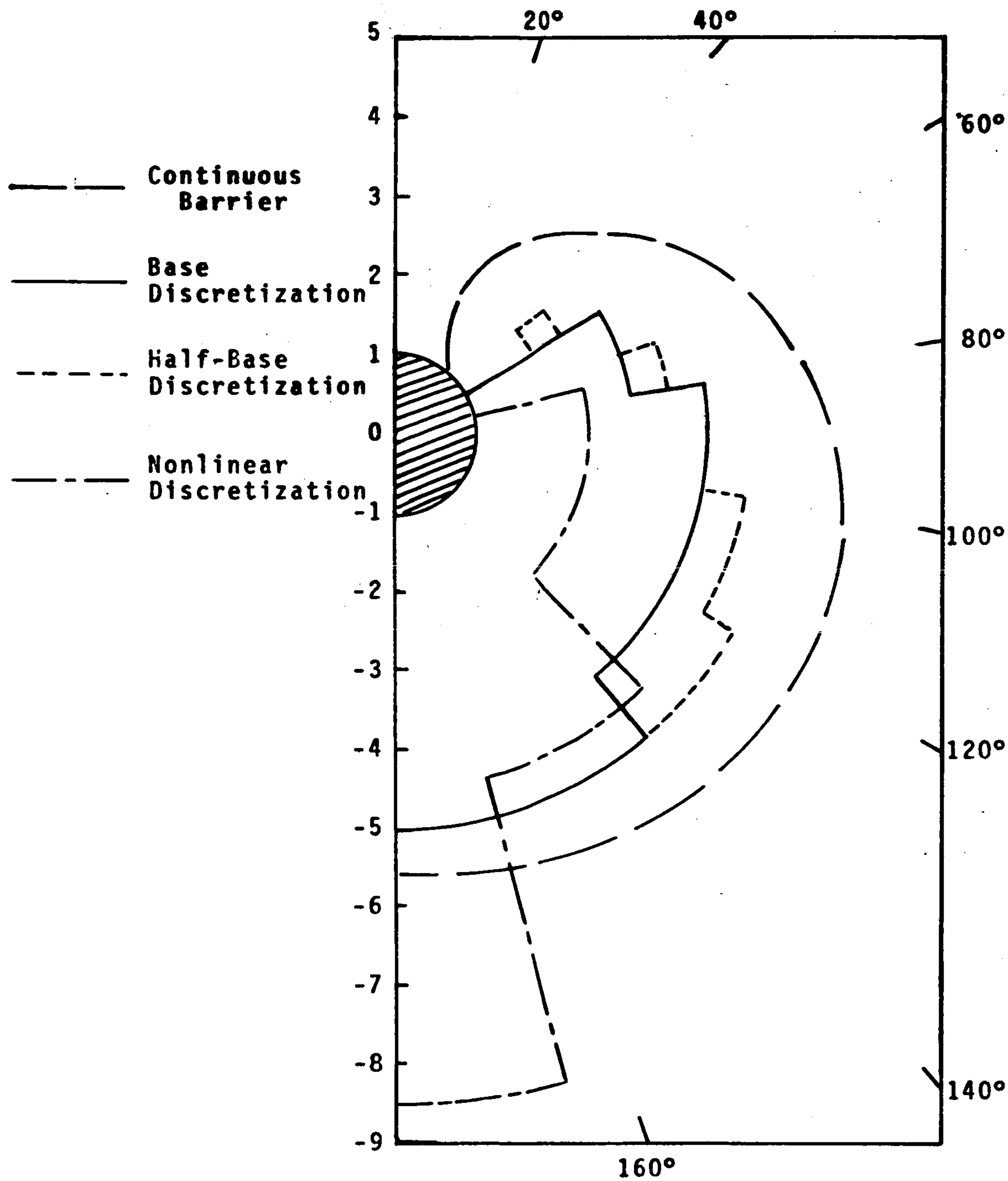

FIG. 3 HOMOCIDAL CHAUFFEUR STRATEGY-BARRIER LOCATION FOR DIFFERENT DISCRETIZATIONS

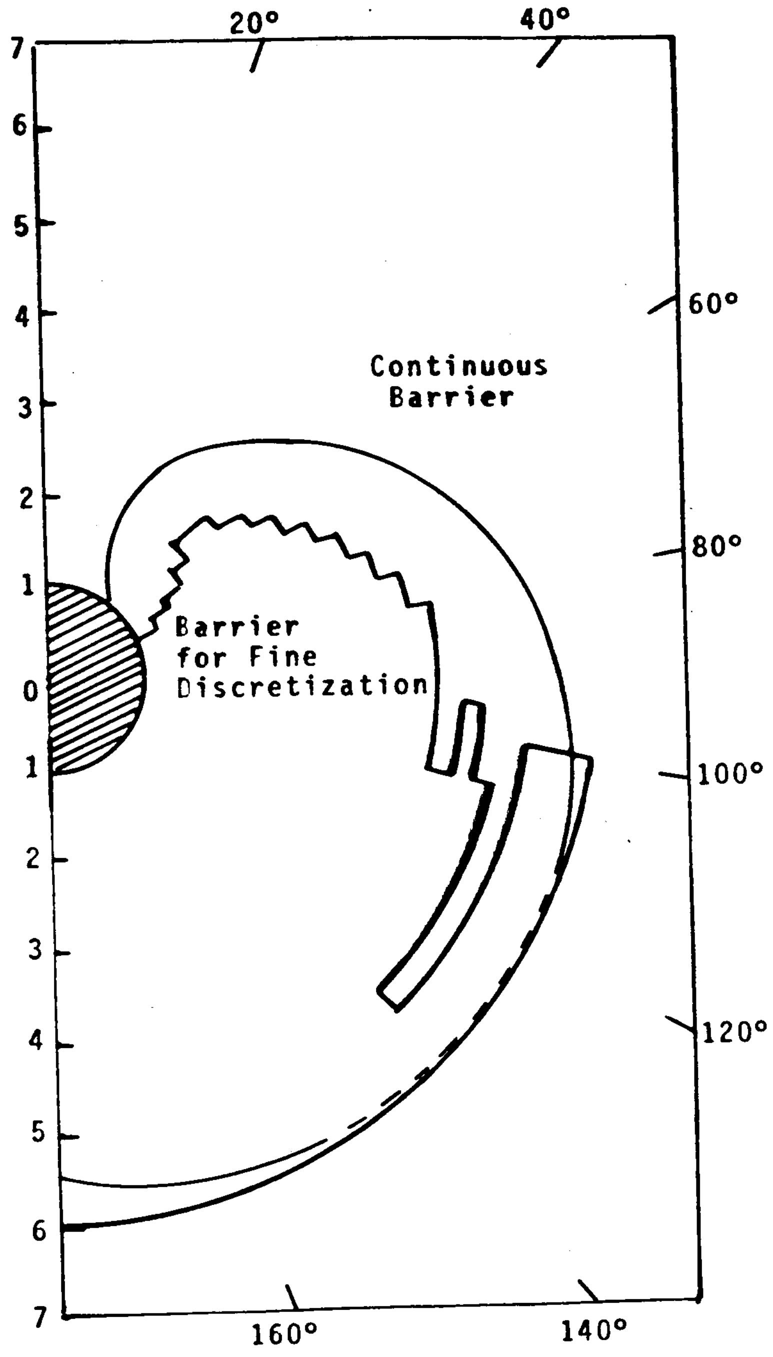

FIG. 4 COMPARISON OF FINE DISCRETIZATION STRATEGY-BARRIER AND CONTINUOUS BARRIER FOR HOMOCIDAL CHAUFFEUR

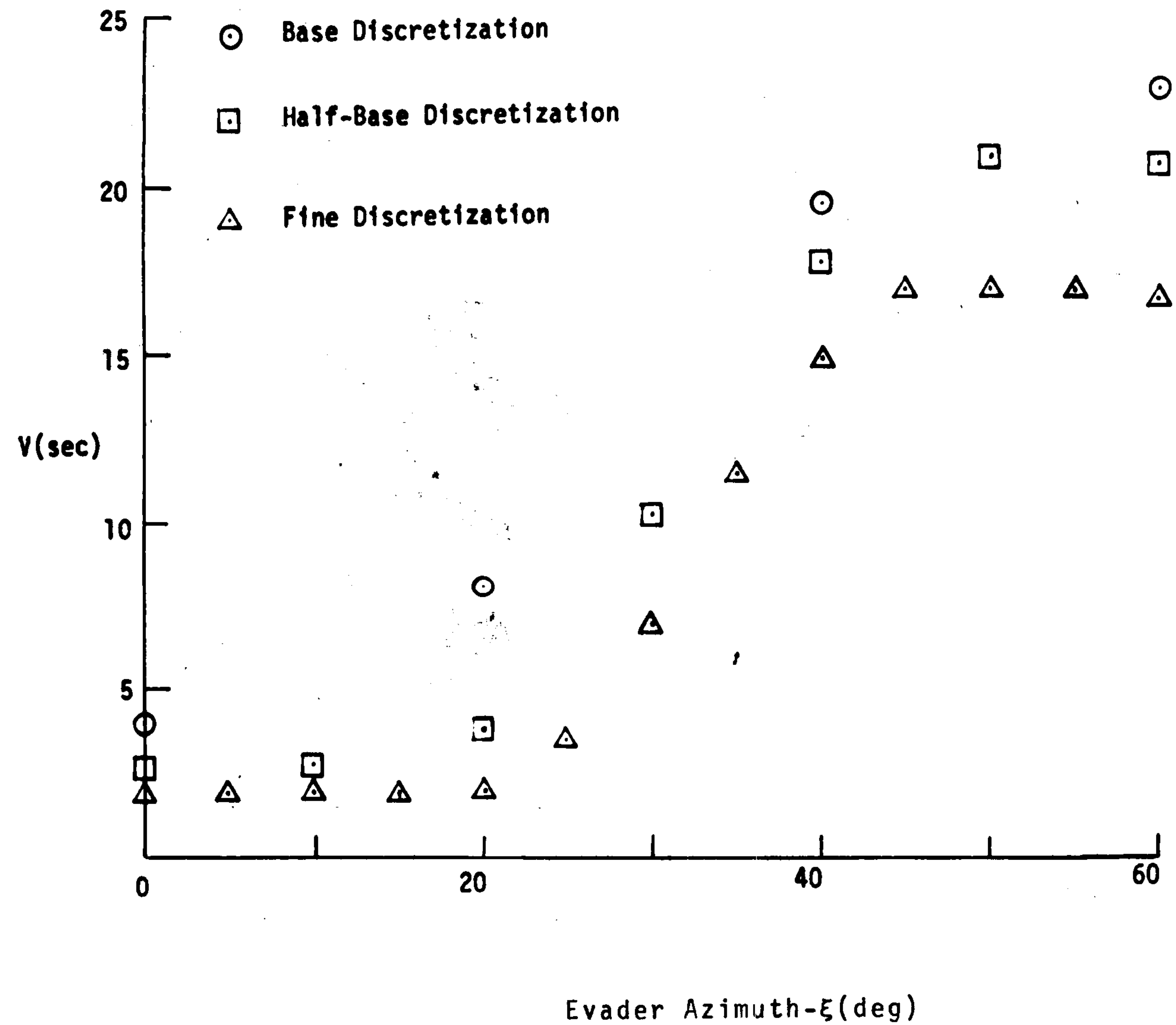

FIG. 5 CHANGES IN OPTIMAL COST (V) NEAR BARRIER FOR DIFFERENT DISCRETIZATIONS

the attacker appears to be penalized more for poor information.

The effect of degraded information is different in the Two-Car problem. A solution was computed for this problem with relative heading, ψ, discretized into a single 360°-block. Such a discretization implies that neither player has any information concerning ψ.* The transition probabilities, and hence the optimal strategies, are then computed with the ψ-variation "averaged out". For this situation, we found that times-to-capture were significantly less than those for the Two-Car problem with ψ discretized into four blocks. Hence, in this problem, it is the evader that is penalized most by the imperfect information. This is easily understood when one notes the relative simplicity of the attacker's strategy; with few exceptions, the optimal strategy for the attacker is to turn right when the evader is to the right, regardless of the value of ψ. Thus, the loss of ψ-information is relatively inconsequential for the attacker. On the other hand, the evader's strategy is quite dependent on ψ and, therefore, he can use better ψ-information to his advantage.

Min-Max and Max-Min

An important question concerning our approach is: "Does the Markov game have a "saddle-point" solution?" This is a difficult question to answer analytically, given the nature of our discretization and the lack of analytic expressions for the p_{ij}'s. Therefore, we decided to explore the question numerically in the Homocidal Chauffeur and Two-Car problems. The option in MAGPIE of computing Min-Max and Max-Min was most helpful here because we test for a saddle-point by comparing Min-Max and Max-Min solutions (by definition, they are equal at a saddle-point). We examine

*This does not reduce the problem to the Homocidal Chauffeur because the evader's turning rate is still limited; it is assumed that both players are aware of this fact.

differences in costs and strategies between Min-Max and Max-Min solutions. In comparing costs, we need only look at the first, non-capture range-block, dead ahead of the attacker, (i.e., at $\{r(2),\xi(1)\}$). This block exhibits the greatest percentage difference in cost; also, inasmuch as most trajectories eventually pass through the block, differences in cost in this block are propagated to the remaining blocks.

For the Homocidal Chauffeur with the "regular discretization" (a., above), we found

$$\min_u \max_v V(r(2),\xi(1)) = 1.4 \max_v \min_u V(r(2),\xi(1))$$

The actual difference in expected time-to-capture was only .6 second, but the percentage change was significant. For this case there were differences in either attacker or evader control in only seven out of eighty state-blocks, i.e., in less than 10% of the blocks. Of the seven blocks, four had differences in evader control, two in attacker control, and one in both controls. An interesting, but not surprising result, was that all differences were in blocks that were adjacent to either the barrier or the capture circle. It is also noteworthy that it was the evader's control that was different near the barrier; in the Min-Max case he directed his velocity vector toward the barrier, whereas in the Max-Min case he chose the opposite direction.

Min-Max and Max-Min solutions were also computed for the Half-discretization version of the Homocidal Chauffeur (b., above). In this case,

$$\min_u \max_v V(r(2),\xi(1)) = 1.18 \max_v \min_u V(r(2),\xi(1)),$$

which corresponded to an actual difference of .13 sec. The Max-Min strategies differed from the Min-Max strategies in only eleven out of the 324 blocks (less than 4%). Again, it was primarily the evader's control that differed and the differences occurred near the barrier or the capture circle. These results indicate that as the discretization becomes finer the solution is approaching a saddle-point. Inasmuch as the Homocidal Chauffeur differential game is known to have a saddle-point, this is further evidence that the solution of the discrete Markov game approaches that of its corresponding differential game as the mesh gets finer.

The results for the Two-Car problem are even more encouraging with respect to a saddle-point solution. For this problem, discretized as described earlier, the costs were virtually identical for the Min-Max and Max-Min solutions. The strategies differed in only seven out of 252 blocks, in spite of the fact that the discretization is relatively "crude" (e.g., cruder than the regular-discretization of the Homocidal Chauffeur).* It seems that Min-Max and Max-Min solutions will converge more rapidly for the Two-Car problem than for the Homocidal Chauffeur. The likely reason for this is that the Two-Car problem is "smoother", because the evader cannot change direction instantaneously. One would expect similar "smooth" behavior in realistic aerial combat problems.

Trajectories

Optimal trajectories for the Homocidal Chauffeur and Two-Car problems were generated to gain insight into the approach; here, we present an illustrative trajectory from each example.

*We don't know where the singular surfaces of the Two-Car problem are but it should be mentioned that the differences in control choices all occur in regions where the choices are changing.

Figure 6 shows a trajectory in the r-ξ plane for the Homocidal Chauffeur problem that starts with the evader in a position below the barrier. The initial part of the trajectory is analogous to what would be obtained in the continuous problem. The attacker first turns away from the evader (a swerve) and then dashes when the target is "behind" him. During this time the evader is "chasing" the attacker. When the separation is great enough (the trajectory is below the discrete barrier, See Fig. 4), the attacker starts turning into the evader, bringing him to a position almost in front ($\xi=20°$) at a distance of about $r=8.5$. At this point, the discrete nature of the problem becomes evident. We see a "chattering" of the trajectory between state-blocks that have different attacker strategies. During this "chatter" the evader is attempting to "flee" along a 45° direction, while the attacker alternates between a turn and a dash. When the range decreases to $r<4$, the attacker's strategy is to turn toward the evader, even when $\xi<20°$. Thus, the target is brought dead-ahead ($\xi=0°$). Another "chatter-trajectory", involving alternating right and left turns for the attacker, brings the target to the capture circle.

The existence of the chatter solution is not surprising, after the fact. It can probably be removed by assuming that the transition between states is not perceived perfectly. For example, if some detection threshold had to be exceeded before a transition was perceived the chattering might disappear. This is worth investigating, even though the chatter effect is probably not too important from a practical viewpoint.

A Two-Car trajectory is shown in Fig. 7. The evader starts out to the right ($\xi=90°$), just outside the attacker's turn-radius ($r\approx3.4$), and headed toward the attacker ($\psi=270°$). This position is very favorable to the evader; it seems he will remain at large

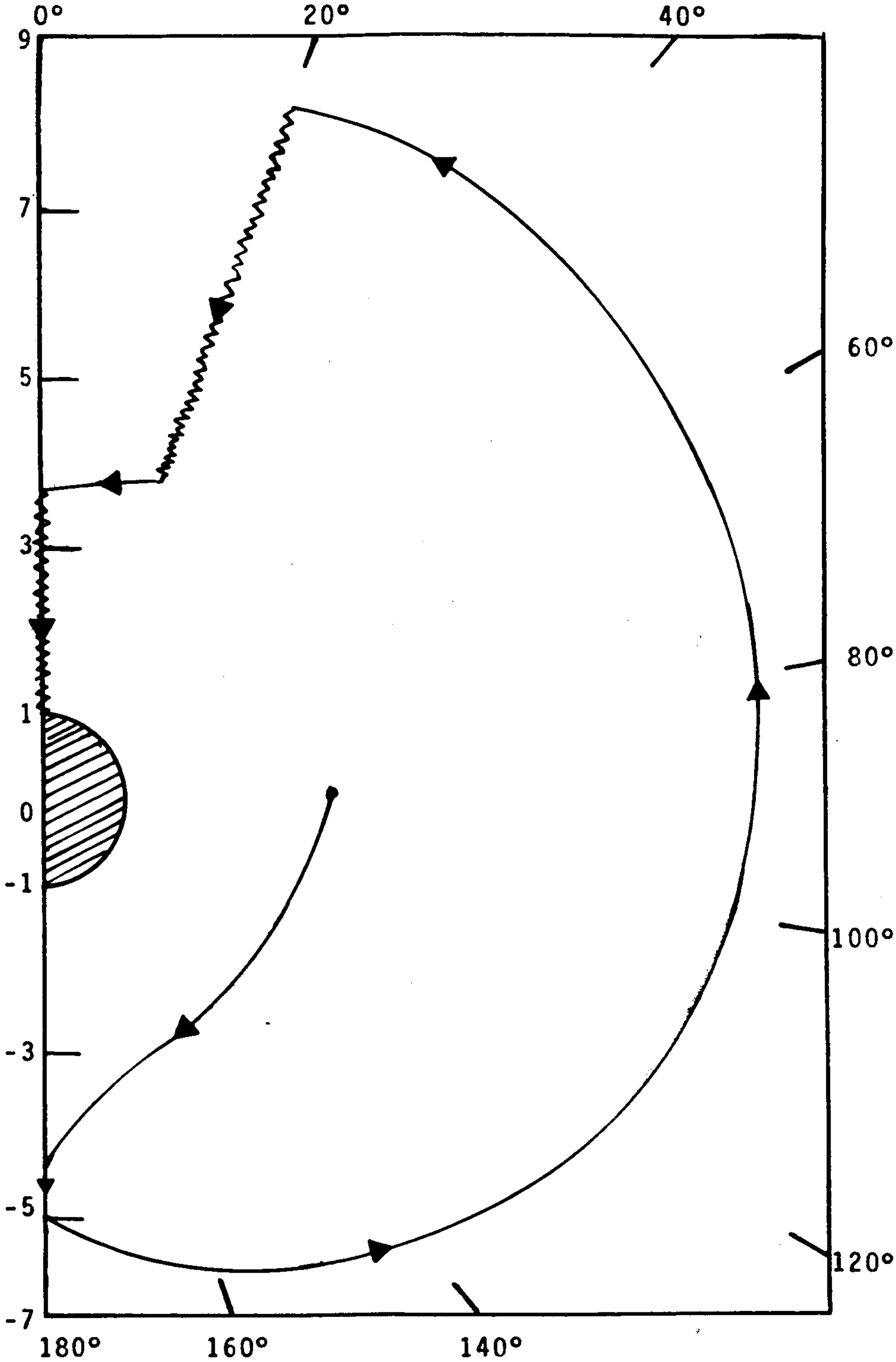

FIG. 6 SWERVE-TRAJECTORY FOR HOMOCIDAL CHAUFFEUR

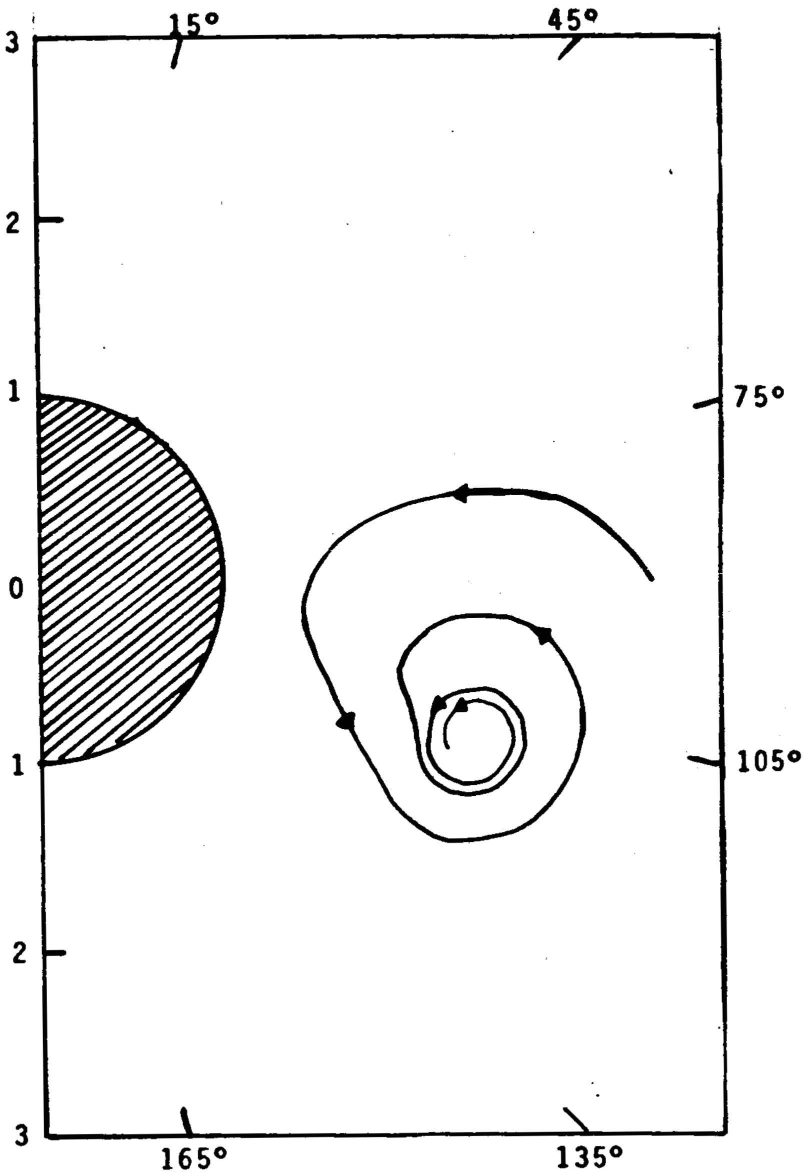

FIG. 7 A NON-CAPTURING, TWO-CAR TRAJECTORY

indefinitely. Actually, the computation was stopped after t=35 seconds (compared to an expected time-to-capture of about 11 seconds). By this time, it appeared that the trajectory was limited to a closed portion of the space and that the probability of capture was virtually zero. How could this happen, when the cost remained finite?

A possible explanation for this behavior is illustrated in Fig. 8. An initial point in the shaded region may yield a closed trajectory in S_5, S_6, S_7, S_8 as shown, for a given control pair, say $(\bar{u},\bar{v})$. However, points in the non-shaded regions of these

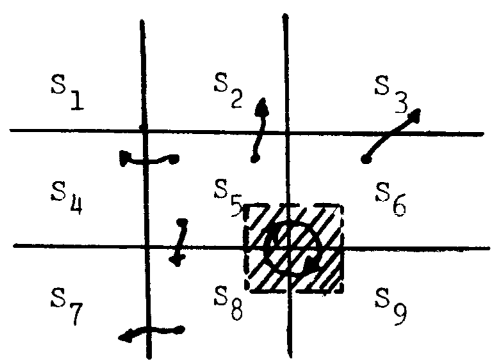

FIG. 8 A CLOSED TRAJECTORY IN DISCRETE STATE SPACE

state-blocks may be driven to different blocks, for the same control choices. Thus, e.g., $p_{52}(\bar{u},\bar{v})\neq 0$ or $p_{63}(\bar{u},\bar{v})\neq 0$. If the other states link to the capture block, there will be a finite probability of capture when $\bar{u},\bar{v}$ is played in S_5 (or S_6, etc.) and the cost will remain finite. Thus, if trajectories behaved in the way the p_{ij}'s are calculated* the "closed" trajectory would not occur

*That is, once a trajectory entered a block, it's position within the block would be selected randomly.

and the evader would be captured eventually. But real trajectories do not act that way.

The above phenomenon is undoubtedly the consequence of solving, in an "average" sense, a game involving poor information. It might even be argued that there is a reasonable chance that such a "closed loop" would occur in some combat situations. However, in a realistic situation, one of the players would undoubtedly "break the chain" after a time. This doesn't happen here because we have solved the steady-state game.

Before leaving this discussion, we note that other Two-Car trajectories did result in capture. Indeed, for the same r,ξ initial conditions, trajectories were computed starting at $\psi=0°$ and $\psi=90°$; in both cases, the evader was captured in less than 12 seconds.

Computational Procedures

The special computational features included in MAGPIE were tested in the Homocidal Chauffeur and Two-Car problems. In particular, we investigated the "regularization" of the integrator time-step and the treatment of "corner" transitions. Solutions were computed for various fixed- and "regularized"-time steps and were compared with those of Reference 1. The solutions in Reference 1 were obtained with the p_{ij}'s calculated from analytic integrals of the equations of motion, evaluated at discrete times; in addition, corner transitions were allowed in Reference 1. No significant differences between solutions were observed for Δt's in the range of .01 to .1, or when the number of steps to leave the block was 5 to 10.

We also experimented with the acceleration feature of the iterator and with the technique of not performing a min-max every step. These techniques worked very well in the Two-Car problem and in the Homocidal Chauffeur problem with the Regular- and Half-Discretizations, i.e., when the number of state-blocks, N, was relatively small. They were not successful for the Fine-Discretization version of the Homocidal Chauffeur or for the planar combat problem described in the next chapter, two problems involving large N. Thus, the utility of the techniques seemed to depend on N. Further study is required to understand and possibly alter this behavior.

5. THE VARIABLE-SPEED PLANAR COMBAT PROBLEM

In this chapter we discuss the application of the Markov game approach to a reasonably complex aerial combat problem. This problem, involving pursuit and evasion at constant altitude, provides a good test of the feasibility of the Markov game approach and is also of intrinsic interest. Here, we describe the problem and its formulation as a discrete Markov game, and present and discuss briefly some typical and interesting "optimal" solutions.

Equations of Motion

We, consider an engagement in which the trajectories of both aircraft remain in the same horizontal plane. Each aircraft can control its linear acceleration by the application or removal of thrust; its rate of turn at a given speed is controlled by the choice of aerodynamic load factor. Thrust, load factor and aerodynamic drag are each nonlinear functions of Mach No. It is interesting to note that both the Homocidal Chauffeur and Two-Car problems are highly idealized versions of this variable speed planar combat problem.

The equations of motion for such an engagement*, in an attacker centered coordinate system that rotates to maintain the x-axis aligned with the attacker's velocity vector** (see Figure 9), may be written as

---

*Basic equations and data for this problem were provided by Martin Moul and David Roberts of National Aeronautics and Space Administration, Langley Research Center.

**A "reduced space" in Isaacs' terminology.

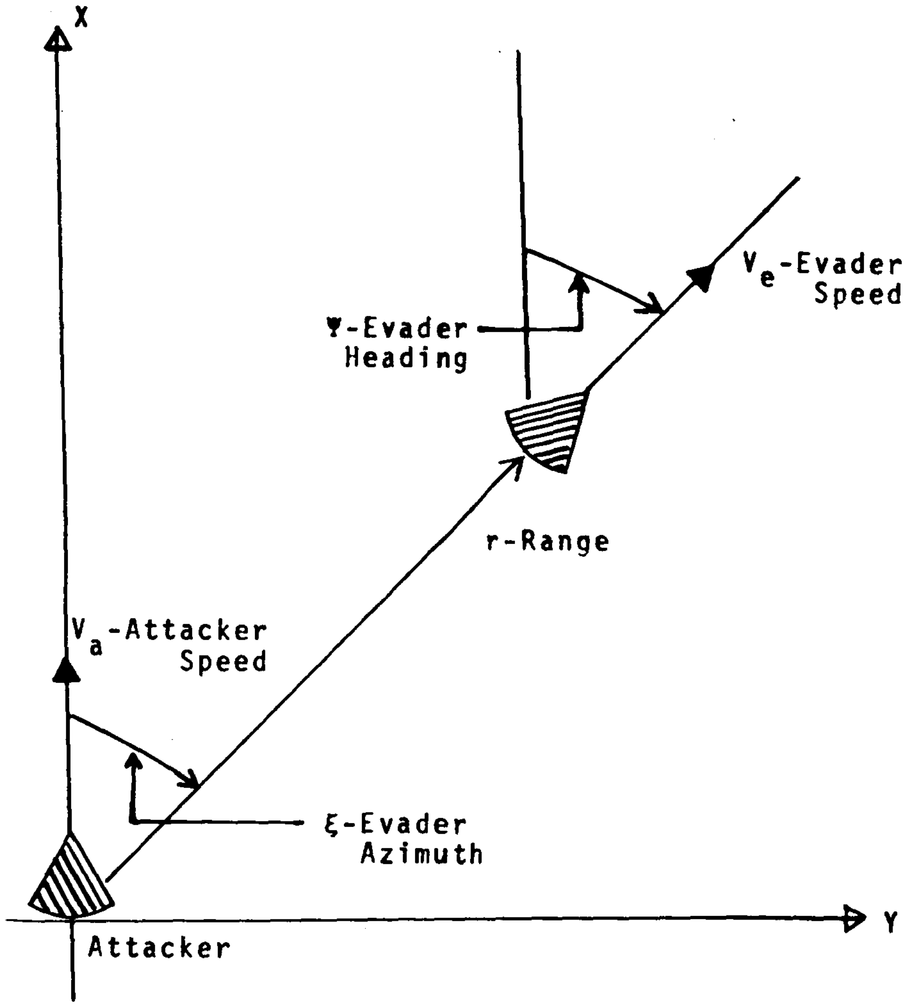

FIG. 9 COORDINATE SYSTEM FOR VARIABLE-SPEED PLANAR COMBAT PROBLEM

$$\dot{r} = a_s[M_e \cos(\psi-\xi) - M_a \cos \xi]$$

$$\dot{\xi} = \frac{-u_1 g\sqrt{n^2_{a_{max}}-1}}{a_s M_a} + \frac{a_s}{r}[M_e \sin(\psi-\xi) - M_a \sin \xi]$$

$$\dot{\psi} = \frac{v_1 g\sqrt{n^2_{e_{max}}-1}}{a_s M_e} - \frac{u_1 g\sqrt{n^2_{a_{max}}-1}}{a_s M_a}$$

$$\dot{M}_a = \frac{g}{a_s}\left[u_2 \frac{T_{a_{max}}}{W_a} - \frac{q_a S_a}{W_a} C_{D_{o_a}} - \frac{W_a}{q_a S_a} C_{D_{C^2_{L_a}}} n^2_a\right]$$

$$\dot{M}_e = \frac{g}{a_s}\left[v_2 \frac{T_{e_{max}}}{W_e} - \frac{q_e S_e}{W_e} C_{D_{o_e}} - \frac{W_e}{q_e S_e} C_{D_{C^2_{L_e}}} n^2_e\right]$$

where the state and control variables are

r : evader's range, ft.

ξ : evader's azimuth, rad

ψ : evader heading (angle between velocity vector of evader and velocity vector of pursuer), rad

M_a: Mach No. of attacker

M_e: Mach No. of evader

u_1: attacker's turning control, $-1 \leq u_1 \leq 1$

u_2: attacker's thrust control, $0 \leq u_2 \leq 1$

v_1: evader's turning control, $-1 \leq v_1 \leq 1$

v_2: evader's thrust control, $0 \leq v_2 \leq 1$

and the remaining symbols are standard nomenclature (see List of Symbols), with the subscripts (a) and (e) referring to attacker and evader, respectively. The attacker and evader aircraft are assumed to have the same basic characteristics. The weight and wing-area are 35,000 lbs and 640 ft^2, respectively. Aerodynamic and thrust data, as a function of Mach No., are given in Table 1. (The thrust data is for an altitude of 15,000 ft.) These data are reasonable for a high-performance fighter aircraft. To allow for differences in attacker and evader performance, we let

$$T_{e_{max}} = \varepsilon_1 T_{a_{max}}$$

$$C_{L_{e_{max}}} = \varepsilon_2 C_{L_{a_{max}}}$$

$$C_{D_{o_e}} = \varepsilon_3 C_{D_{o_a}}$$

$$C_{D_{C^2_{L_e}}} = \varepsilon_4 C_{D_{C^2_{L_a}}}$$

The ε_i's were constants, free to be selected at computer run-time. For the example discussed here, ε_1=.7, $\varepsilon_2=\varepsilon_3=\varepsilon_4$=1.0; thus, the only difference between the two aircraft was that the evader's maximum thrust was 70% that of the attacker's, at all speeds.

The maximum normal acceleration for each aircraft was assumed to be 6g. However, because of a constraint on the maximum lift, flight conditions occur for which it is not possible for the aircraft to pull 6g's; for those conditions, the maximum normal acceleration is reduced accordingly. It is worth noting that only the component of normal acceleration that is in the horizontal plane contributes to the turn.

TABLE 1

Aerodynamic and Thrust Data

M	$C_{L_{max}}$	C_{D_o}	$C_{D_{C_L^2}}$	$10^{-4}T_{max}$ (lbs)
0.4	1.0	.013	.295	2.03
0.6	1.0	.013	.295	2.27
0.8	1.0	.013	.295	2.46
1.0	1.1	.025	.310	2.75
1.2	1.1	.027	.330	3.01
1.4	.9	.025	.380	3.34
1.6	.7	.023	.440	3.64
1.8	.5	.020	.500	4.00
2.0	.5	.020	.560	4.54

The Markov Game

We define the finite, or "compactified", "playing"-space for this problem by

$$X=\left\{(r,\xi,\psi,M_a,M_e):\ 100'\leq r\leq 15{,}500',0\leq\xi\leq 2\pi,0\leq\psi\leq 2\pi,.5\leq M_a\leq 1.75,.5\leq M_e\leq 1.5\right\}$$

The limits on r, M_a and M_e, serve to bound the region of the state-space that is of interest here;* these boundaries were implemented as "reflecting barriers" (See Ref.1).

*The lower limit on range was introduced to avoid the singularity at the origin of the polar coordinate system.

The state space was discretized in the manner indicated in Table 2. Range was discretized in nonlinear fashion to reflect

TABLE 2

Table of State-Block Interfaces

Block No.	r(ft)	ξ(deg)	ψ(deg)	M_a	M_e
1	100.	-15.	45.	.500	.500
2	500.	15.	135.	.750	.750
3	1500.	45.	225.	1.000	1.000
4	3500.	75.	315.	1.250	1.250
5	7500.	105.	405.	1.500	1.500
6	15500	135.		1.750	
7		165.			
8		195.			
9		225.			
10		255.			
11		285.			
12		315.			
13		345.			

the fact that resolving range accurately becomes more difficult and less important as range increases. The discretization of the azimuth and relative-heading angles was identical to that used in the Homocidal Chauffeur and Two-Car problems and was based on the same rationale employed there. (Figure 10 illustrates the discretization of the relative position (r,ξ) and shows the attacker's maximum turning capability at various speeds.) The Mach-states, M_a and M_e, were discretized linearly into blocks of .25-Mach No.; these Mach No. discretizations seemed reasonable, though they were chosen somewhat arbitrarily.

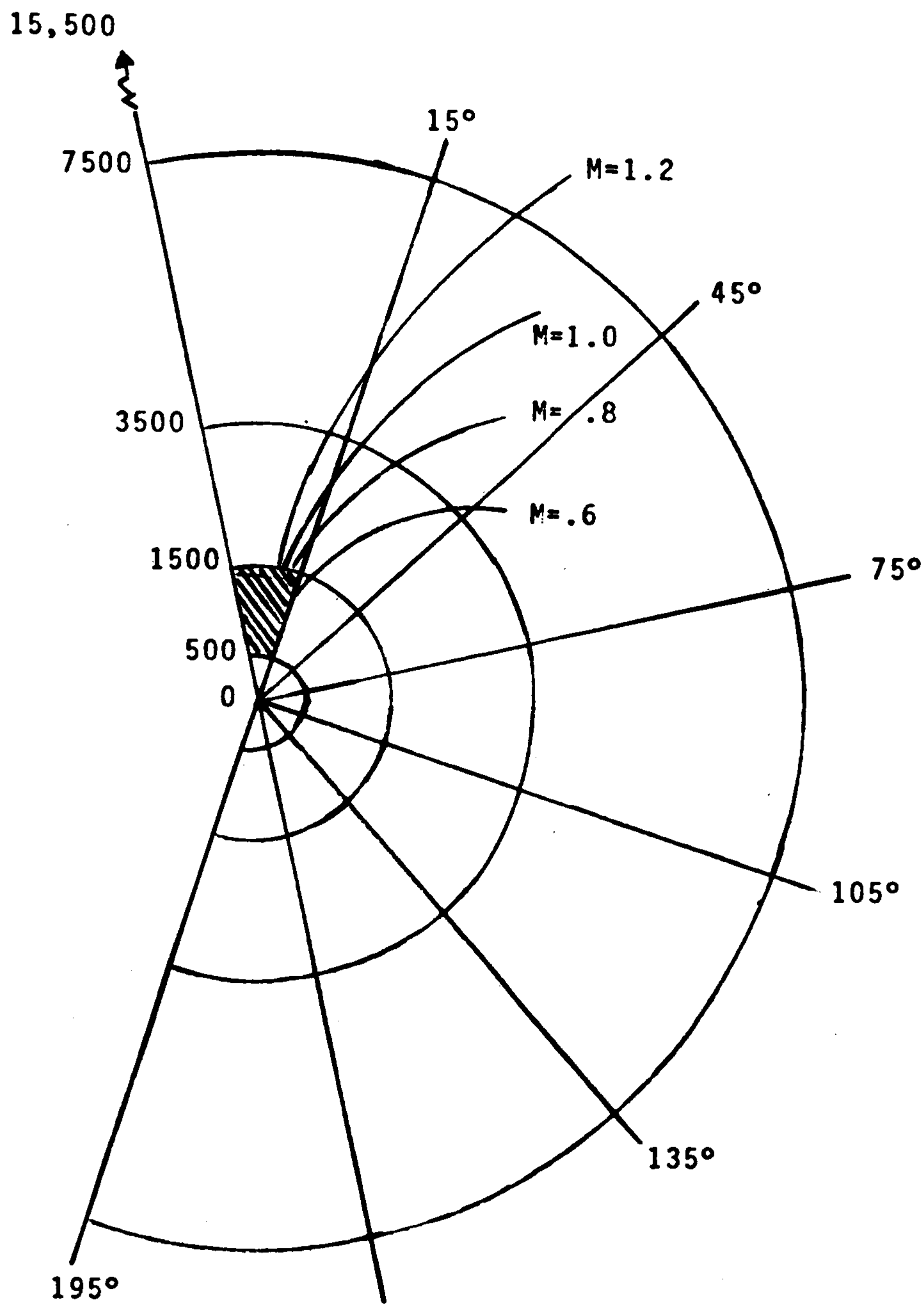

FIG. 10 GEOMETRY OF RELATIVE POSITION DISCRETIZATION

The total number of state blocks for the problem is 4800. However, symmetry with respect to azimuth and relative heading can be employed so that for computation purposes we have, effectively, N=2800.

In the continuous formulation of the problem each aircraft could choose any fraction of maximum load factor (plus or minus) and of maximum thrust. For the discrete formulation, we allowed the following choices:

$$u_1, v_1 = -1, 0, +1$$

$$u_2, v_2 = 0, 1$$

Thus, each aircraft could make a max g-turn in either direction or a dash, while employing maximum- or zero-thrust. This meant that there were six possible maneuvers for each player.

Capture was defined by the condition $\{100' \leq r \leq 1500', -15° \leq \xi \leq 15°\}$. This capture condition was somewhat unrealistic insofar as no requirements were placed on relative heading or on time in the capture region. It is possible to include such capture requirements within the MAGPIE framework. However, in this example where we have assumed near equality in performance of the competing aircraft, capture would then be virtually impossible. At this stage in development of the Markov approach, we believed our understanding would be better served by solving the problem with the less realistic capture condition.*

*Actually, we tried a more meaningful criterion for capture on a test problem in which the state-space discretization was much cruder. We found the time-to-capture was extremely large, and the iterative solution had still not converged!

The incremental cost, $c_{ij}(\underline{u},\underline{v})$ was taken to be the average time of a transition from S_i to S_j using maneuver pair $(\underline{u},\underline{v})$. The total cost is then the expected time-to-capture; the attacker attempts to minimize this time, the evader to maximize it.

Results

The complete "solution" to the variable speed planar combat problem comprises a considerable amount of data. Inasmuch as we are not concerned here with the details of the solution, we abstract the data in the hope of providing an insight into the nature of the results for this complex Markov game. To this end, we give brief consideration to optimal-costs, -strategies and -trajectories.

Optimal costs: The cost for each state-block is the expected time-to-capture for an encounter *starting* somewhere within the five-dimensional hypercube defining that block. The optimal cost is the cost obtained when both players use their optimal strategies. Before discussing specific cost-results, we note that because of the uncertainty in the problem, the optimal costs are only indicators; numerical values of the cost have little physical significance.

In Figure 11, we present the optimal costs for engagements starting with a range between 7500' and 15,500' and the evader's Mach No. being somewhere between 1.0 and 1.25.* These results agree well with intuition. For the most part, cost increases with increasing azimuth angle, for a given attacker speed. When the evader is "straight-ahead" ($\xi(1)$), cost decreases with increasing attacker speed, except when the evader is heading toward

*In this, and subsequent figures, the notation x(i) refers to the i-th block of the state variable x; e.g., r(5) and $\xi(3)$ refer to the fifth range-block and the third azimuth block (See Table 2).

$\xi(\cdot)$: Evader-Azimuth Cell, $\psi(\cdot)$: Evader Heading Cell, $M_A(\cdot)$: Attacker Mach No. Cell

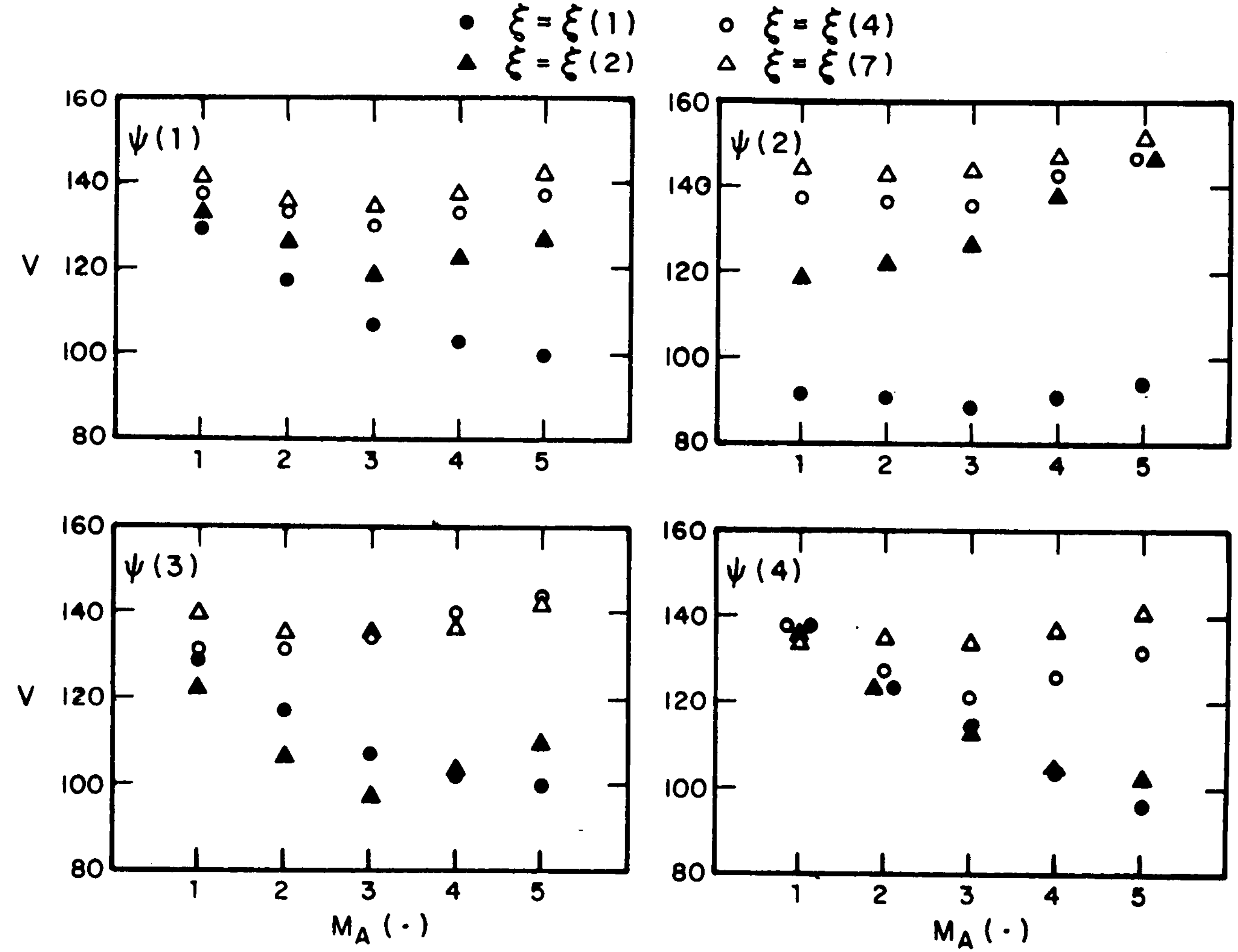

FIG. 11 VARIATION IN EXPECTED TIME-TO-CAPTURE, V, WITH ATTACKER INITIAL SPEED ($7500' \leq r_o(5) \leq 15{,}500'$, $1.0 \leq M_{e_o}(3) \leq 1.25$)

the attacker ($\psi(2)$). If the evader is off to the right or behind the attacker, then it appears that the attacker is generally better off if his speed is "equal to" or slower than that of the evader.

Figure 12 gives results for starting conditions in which the speeds of the attacker and evader are approximately equal. Only the intermediate and furthest range-blocks are considered; attention is also restricted to the cases where the evader is initially headed to the left ($\psi(3)$) or is travelling in about the same direction ($\psi(4)$). We see that costs increase uniformly with speed for a given azimuth angle and that they generally increase with azimuth angle for a given speed. Perhaps the most interesting aspect of these results, is the suggestion of a cost-barrier, particularly for $\psi(3)$ cases.

Optimal strategies: Table 3 presents the optimal attacker strategies when both aircraft are at approximately the same speed, i.e., $.75 \leq M_a, M_e \leq 1.0$.* As far as turning strategy is concerned, we see that when the evader is headed to the right ($\psi(1)$) or is travelling in the same direction as the attacker ($\psi(4)$), the pursuit strategy is essentially to turn toward the evader. When the attacker and evader are headed in opposite directions ($\psi(2)$), there is a region in-tight and "off-to-the-side", where it is optimal to turn away (a swerve). The most interesting situation occurs when the evader is headed to the left ($\psi(3)$).(See Fig.13.) When the evader is at 12 o'clock ($\xi(1)$) the attacker should turn left, presumably in an effort to get into a tail chase. At 1 o'clock the optimal strategy is to turn into the evader when he

*Table entries give turning control then thrust control; thus RO means turn right with thrust off.

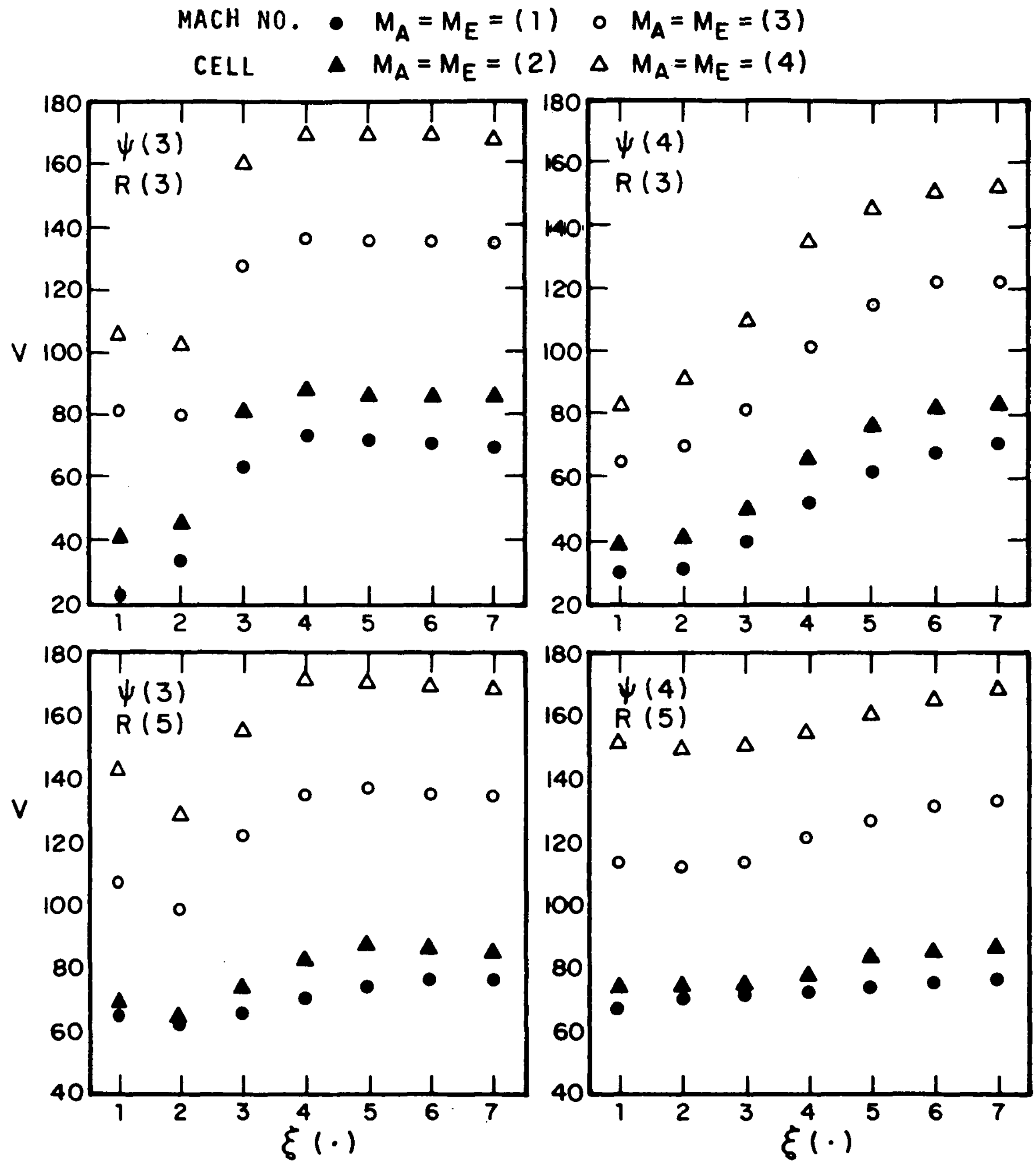

FIG. 12 VARIATION IN EXPECTED TIME-TO-CAPTURE, V, WITH AZIMUTH CELL, ξ(·), FOR DIFFERENT MACH NO. REGIMES (R(·): RANGE CELL, ψ(·): HEADING CELL)

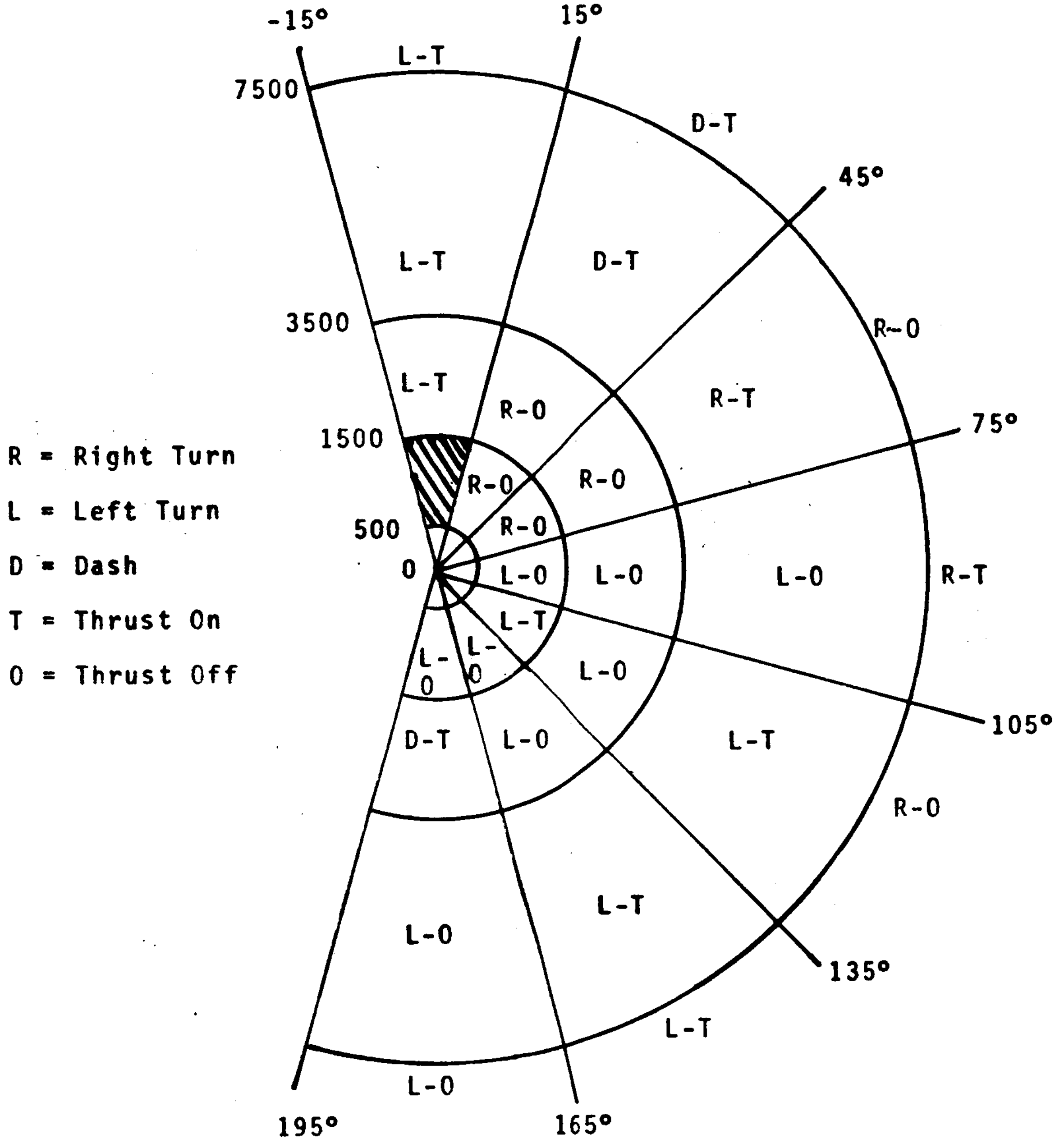

FIG. 13 OPTIMAL PURSUIT STRATEGY FOR: $225° \leq \psi(3) \leq 315°$; $.75 \leq M_A(2)$, $M_E(2) \leq 1.00$.

TABLE 3

Optimal Pursuit Strategies for $.75 \leq M_a, M_e \leq 1.0$

L=Left turn, R=Right turn, D=Dash, T=Thrust On, 0=Thrust Off

		ξ(1)	ξ(2)	ξ(3)	ξ(4)	ξ(5)	ξ(6)	ξ(7)
ψ(1)								
	r(1)	LO	LT	DO	RT	RT	RO	RO
	r(2)	--	LO	RT	RT	DO	RT	RO
	r(3)	RT	RO	RO	RO	RT	DO	DT
	r(4)	RT	RT	RT	RO	RT	RT	RO
	r(5)	RT	RT	RT	RT	RT	RT	RO
ψ(2)								
	r(1)	DO	LO	LT	LT	DO	RT	RO
	r(2)	--	LO	LT	LO	RT	RT	RT
	r(3)	DO	RO	LT	LT	RO	RT	DO
	r(4)	DT	RO	RO	RT	RT	RO	RO
	r(5)	DT	RT	RO	RO	RO	RO	RT
ψ(3)								
	r(1)	RO	RT	RT	LO	LO	LO	LO
	r(2)	--	RO	RO	LO	LT	LT	LO
	r(3)	LT	RO	RO	LO	LO	LO	DT
	r(4)	LT	DT	RT	LO	LT	LT	LO
	r(5)	LT	DT	RO	RT	RO	LT	LO
ψ(4)								
	r(1)	DO	RO	RO	RO	LO	LO	RT
	r(2)	--	RO	RO	RO	RO	RO	LO
	r(3)	DT	RT	RT	RT	RO	RO	RO
	r(4)	RO	RT	RT	RT	RO	RO	RO
	r(5)	LT	RT	RT	RT	RT	RT	RT

is near and to "close" with a dash at greater distances. It is still optimal to turn into the evader when he is at 2 o'clock. However, the situation changes when the evader is directly to the right (3 o'clock) or "behind". Then, except at the greatest ranges ($r \geq 7500'$) the evader can get inside the attacker's turn radius, thus forcing the attacker to turn away. Examination of the evader's strategy for this case reveals that, where possible, the evader will attempt to turn into and get behind the attacker; at greater ranges where such a tactic is not likely to be successful, the evasive strategy is to "flee" (see Fig.14).

The optimal thrusting strategy for the pursuer is somewhat more complex although the trends agree with intuition. Generally speaking, when the evader is close-in and tighter turns are required, it is optimal to use zero thrust; at longer range, where it is important to reduce separation, it is optimal to thrust. However, there are many exceptions to these trends. One might be able to explain these exceptions after the fact but it would have been virtually impossible to have forecast them.

Strategies in remaining portions of the state-space were not unlike those described above, with modifications due to changes in speed being about as expected. Thus, at higher attacker speeds the greater turning radii resulted in larger regions in which it was optimal to turn away from the target. Conversely, when the attacker was slower than the evader, the regions in which direct pursuit was appropriate expanded. As for thrust control, the trends were reasonably intuitive but, as above, the particulars were quite unpredictable.

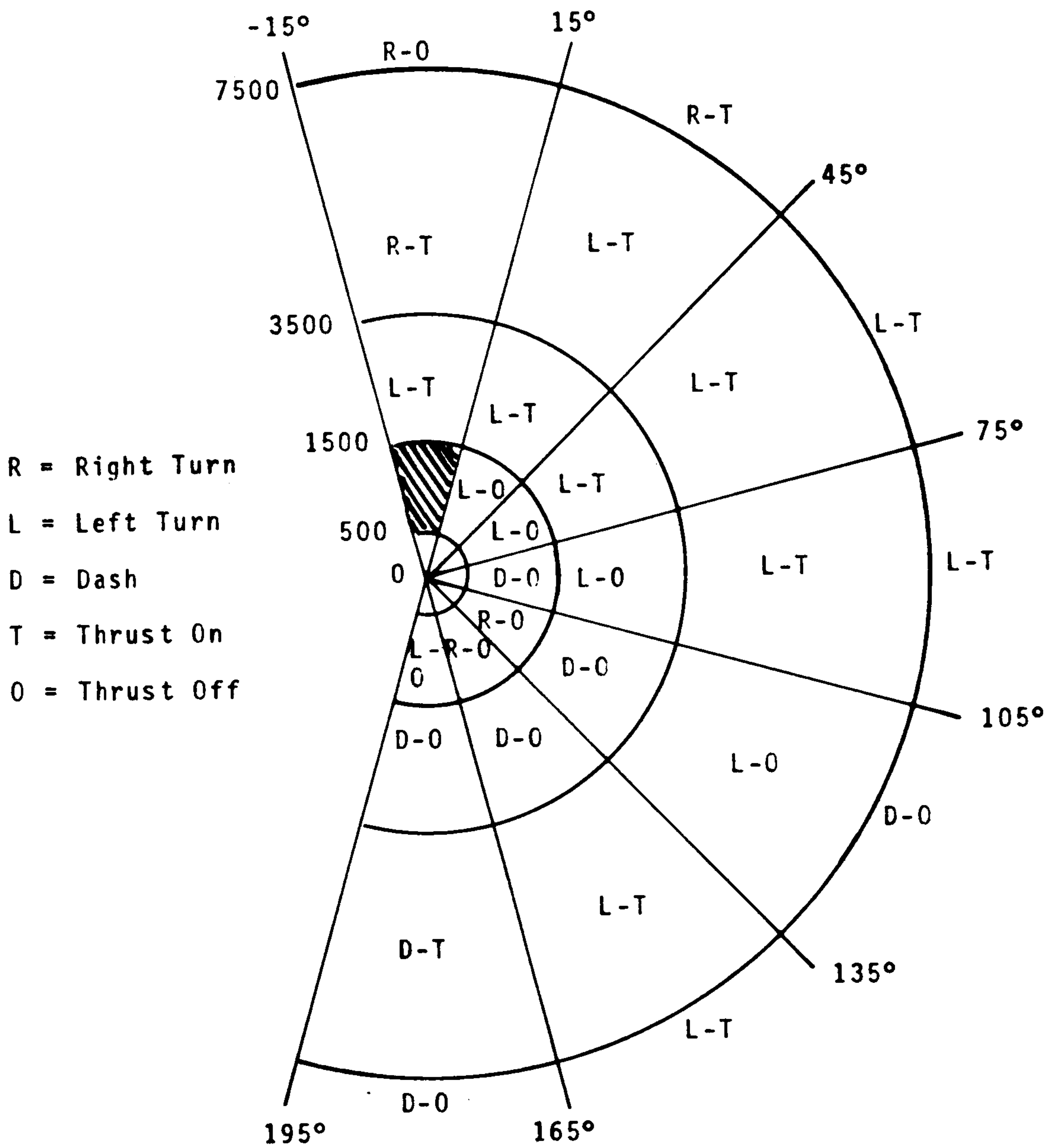

FIG. 14 OPTIMAL EVASIVE STRATEGY FOR: $225° \leq \psi(3) \leq 315°$; $.75 \leq M_A(2), M_E(2) \leq 1.00$.

Optimal trajectories: As mentioned earlier, MAGPIE can be used to generate specific trajectories that result from starting at fixed initial conditions and "playing" the optimal strategies. In this section, we present four such trajectories for the variable speed planar combat problem. The results are plotted in inertial coordinates and we assume that the attacker is initially at the origin.

Figure 15 gives the result of an encounter that starts with evader at fairly long range, off to the right, and headed across the attacker's projected path. (Exact initial conditions for the engagement are shown on the figure, as is the case in subsequent examples.) Initially, the evader executes a minimum radius turn to the right, bringing his Mach No. down to the minimum value of $M_e = .5$ in the process. After turning his flight path by about 115° he begins a nonthrusting dash. This zero thrust choice would appear to be the penalty for imperfect information; so far as the evader (or attacker) is concerned the range is still greater than 3500 ft. and the speeds of the two aircraft are the "same" (all that is known is that $.5 \leq M_a, M_e \leq .75$). When the range becomes less than 3500 ft., the evader starts to apply thrust again in an attempt to escape. The attacker begins the engagement by dashing. While the evader is headed across his path the attacker uses zero thrust. When the two aircraft are heading in the "same" direction, the attacker applies full thrust increasing his speed until $M_a > .75$. He then executes a slight turn to place the target at "12 o'clock", losing speed in the process. At this point, the attacker's strategy duplicates that of the evader, i.e., he dashes using zero thrust until $r \leq 3500$ ft. Thus, the final stage of the engagement is a tail-chase.

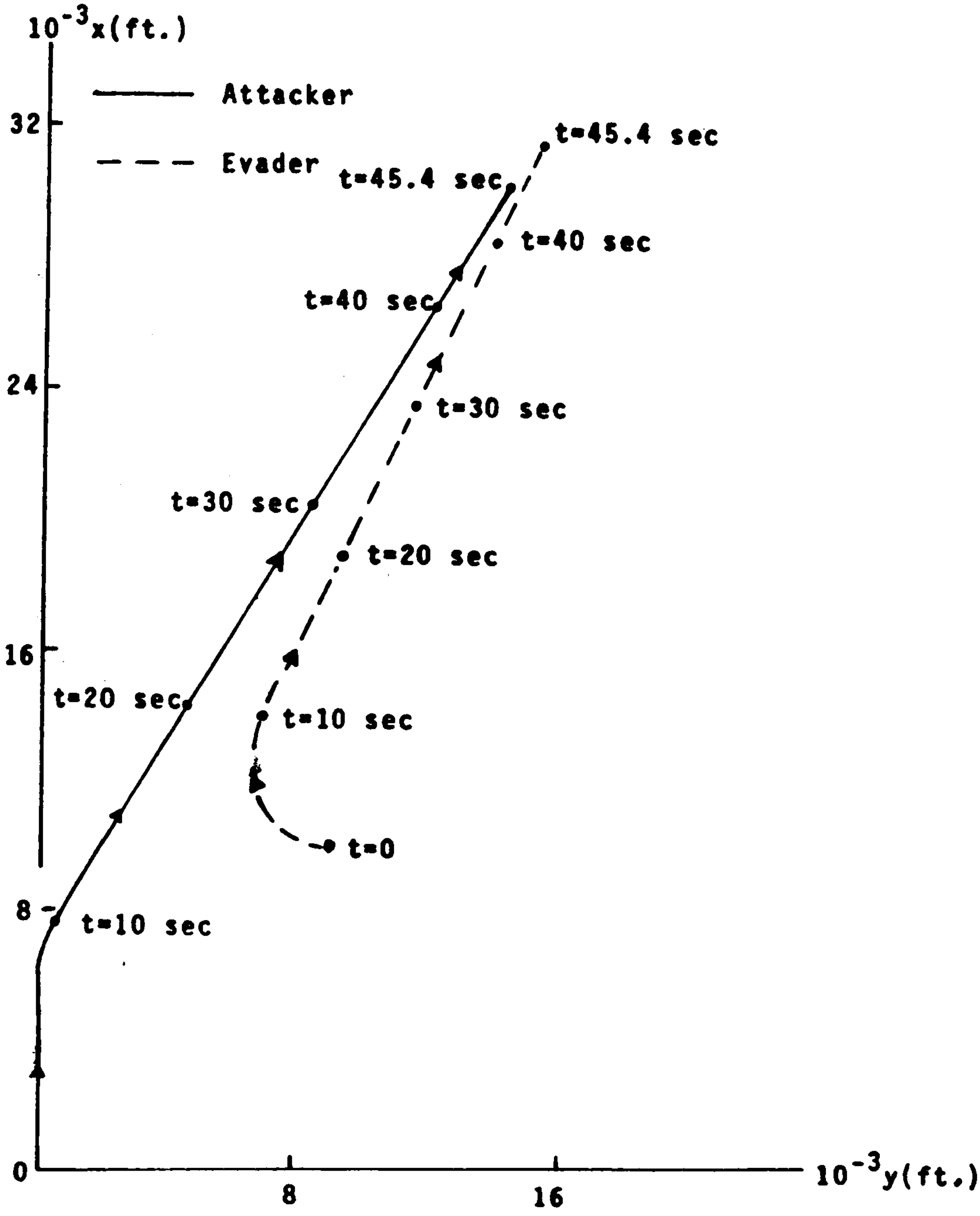

FIG. 15 CAPTURE TRAJECTORY FOR EVADER INITIALLY HEADED ACROSS ATTACKER'S PATH

(r_o=13,500 ft., ξ_o=42°, ψ_o=270°, M_{a_o}=.72, M_{e_o}=.59)

Figure 16 shows an engagement that starts off in somewhat similar fashion except that the two vehicles are headed in approximately the same direction. Actually, the trajectory of the previous example passed through the initial state-block (i.e., r(5), $\xi(2)$, $\psi(4)$, $M_a(1)$, $M_e(1)$) for this case. It can be seen that the basic character of the engagement is the same as the previous one, ending in a tail-chase. It is interesting to note that the final direction in inertial space is quite different. Another interesting point is the continued curving of the flight paths. This arises from a "chattering" along the ξ=15° ray; for the conditions encountered here this ray separates a dash region from a turn region, for each vehicle.

Figure 17 demonstrates a situation in which the evader starts out well within the turn-radius of the attacker, but is headed away (to the right). The evader initially turns right but switches to a dash as his azimuth angle relative to the attacker increases. During this time, range increases as the attacker turns toward the evader. After about six seconds, the two vehicles are headed in approximately the same direction and the evader attempts to turn inside the attacker. The attacker responds first with a dash and then turns with the evader until capture. There is no tail-chase in this engagement!

The final, and perhaps most interesting, trajectory we shall discuss here is presented in Figure 18 . The engagement starts with the evader inside the attacker's turn radius and headed toward him. Moreover, the attacker's speed is about 20% greater than that of the evader. Thus, the evader is in a most advantageous position, as can be seen from the result; namely, capture has not occurred within 90 seconds, and it seems likely that the evader can remain at large indefinitely. The evasive maneuver involves turning into

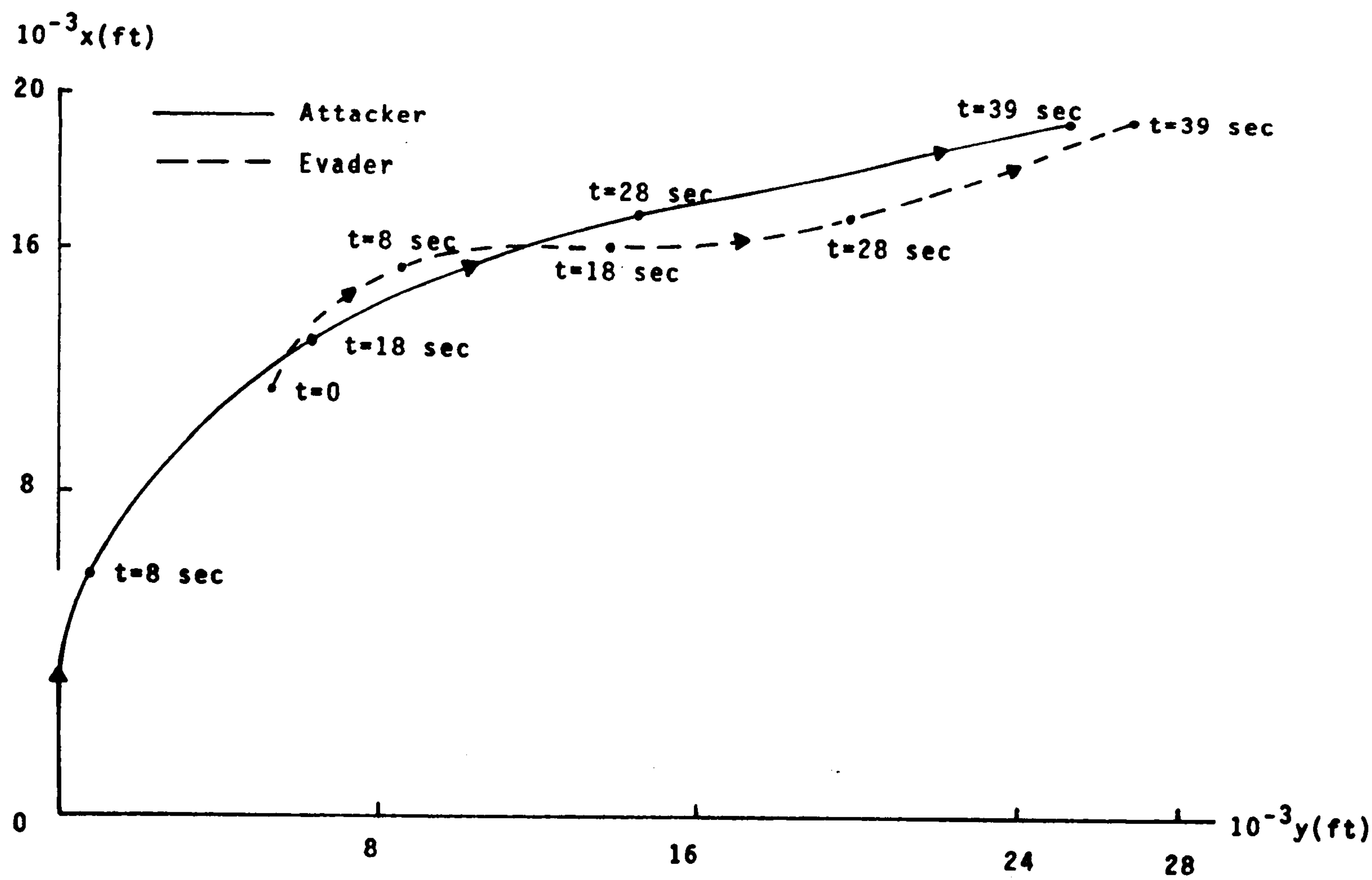

FIG. 16 TRAJECTORY STARTING AND ENDING IN A CHASE
(r_o=11,900 ft., ξ_o=27.6°, ψ_o=6°, M_{a_o}=.67, M_{e_o}=.59)

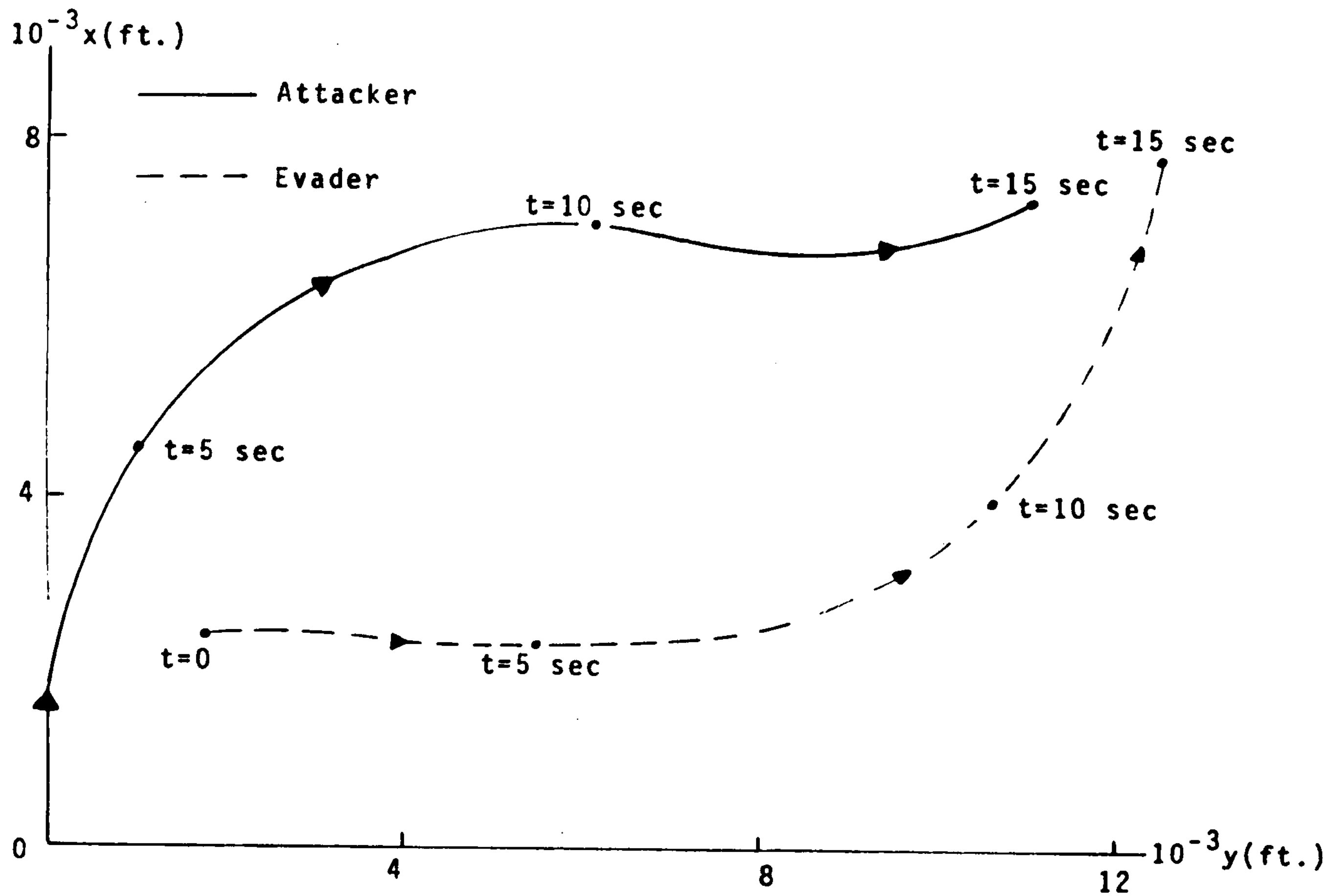

FIG. 17 A "COLLISION-TYPE" CAPTURE TRAJECTORY
(r_o=3000 ft., ξ_o=35°, ψ_o=90°, M_{a_o}=1.15, M_{e_o}=.9)

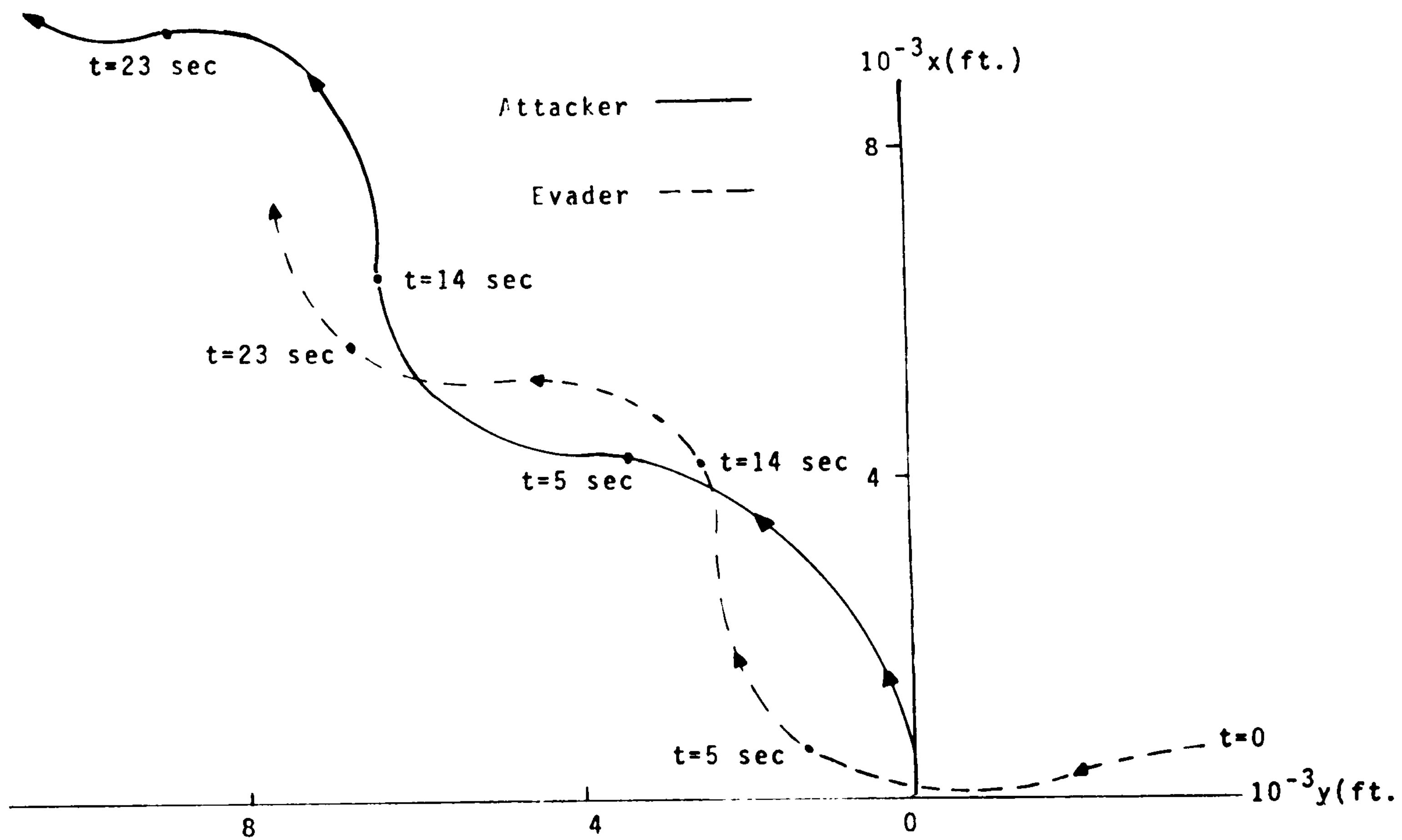

FIG. 18 A NON-CAPTURE TRAJECTORY INVOLVING "SCISSORING MANEUVERS"
(r_o=3600 ft., ξ_o=80°, ψ_o=270°, M_{a_o}=.9, M_{e_o}=.76)

the attacker, forcing him to overshoot in turns because of his greater speed. The attacker, on the other hand, keeps trying to turn away from the evader, so as to obtain sufficient separation for an attack. The resulting paths are highly reminiscent of those characteristic of the well-known "scissors" maneuver.

6. COMPUTATIONAL FEASIBILITY OF MARKOV APPROACH

The major question with respect to the Markov game approach to aerial combat problems was: "Could solutions to significant problems be obtained using acceptable amounts of computer resources?" Our study of the planar combat problem suggests that <u>it is feasible to solve meaningful problems in this way</u>. It also provides data that allows us to examine the limits of our current capabilities. In this chapter, we summarize that data and discuss its implications.

We anticipated that the practicality of the method would hinge on the requirements for high-speed storage and for CPU time. Our experience now indicates that high-speed storage is not a significant problem. The "state-increment" formulation and the elimination of "corner" transitions help to reduce storage requirements to manageable levels. Indeed, the "core" requirements for the five-state, planar combat problem (PCP) were such that this problem was run in standard fashion with other "jobs". In addition, the use of word-packing for storing the p_{ij}'s reduces significantly the time needed for reading from other storage devices.

The situation with respect to CPU time is not so favorable. This can be seen by extrapolating the results of the three problems solved in this study; namely, the Homocidal Chauffeur problem, the Two-Car problem and the Variable-Speed Planar Combat problem (PCP). These problems are similar in nature, but require different amounts of CPU time, primarily because of differences in their respective state and control spaces.

The major portion of the required CPU time (approximately 2/3) is devoted to computing the transition probabilities (p_{ij}'s). The prime factors effecting this computation time are: the number of state blocks (N); the number of control-pair possibilities (UxV); the number of points per block (P) and the average number of integration steps per block (T) used in calculating the relative frequencies identified with the p_{ij}'s. In Table 4, we present values of $C = N \cdot U \cdot V \cdot P \cdot T$ for the three problems being considered along with actual CPU times (CDC-6600) required to solve each of the problems using MAGPIE (i.e., to compute the p_{ij}'s and the optimal strategies). It can be seen from this table that, for timing estimates, it is reasonable to assume that total CPU time is proportional to C.

TABLE 4

Measured CPU Time VS. Problem Size

Problem	$C = N \cdot U \cdot V \cdot P \cdot T$	CPU Time
Homocidal Chauffeur*	1.512×10^5	62 sec
Two-Car	15.75×10^5	530 sec
Aerial Combat	816.5×10^5	≈32,000 sec

Now let us extrapolate these results to a reasonably extensive, three-dimensional air combat problem (3D). The state equations for such a problem may be written in an attacker-centered coordinate system that has its x-y plane horizontal and its x-axis aligned with the horizontal projection of the attacker's velocity vector. We can then take as state variables: target range, azimuth and elevation (r, ξ, η, respectively); relative-heading (ψ) and Mach No. ($\Delta M = M_a - M_e$); attacker altitude (h_a), Mach No. (M_a) and flight path angle (γ_a); and the target flight path angle (γ_e).**

*We use the "crude-discretization" formulation for our estimates.

**Other choices are possible, of course.

Table 5 gives a list of these states along with a block discretization level that appears reasonable in view of pilot capabilities. Also shown in this table are state discretizations for a problem restricted to a vertical plane (VP).

TABLE 5

State	No. of Blocks Per State	
	3-Dimensional	Vertical Plane
r	6	6
ξ	7	--
η*	6	12
ψ	4	--
M_a	3	3
ΔM	3	3
γa*	5	5
γe*	5	5
h_a	4	4
N = Total No. of Blocks	907,200	64,800

*One need only consider $\eta(\gamma)$ in the interval $(-\pi/2, \pi/2)$, because values in the remaining two quadrants correspond to changes in azimuth (heading).

The range, azimuth and relative heading discretizations are the same as those used for the PCP, except that an extra range block (from 15,500 ft.-31,500 ft.) has been added. Elevation angle is given in "clock angles." The three discretization levels for ΔM correspond to the target being "faster", "slower" or at "approximately the same speed" as the attacker. The three attacker Mach "blocks" should give adequate speed range for an engagement. Flight path angles are assumed to be discretized in 36° increments, corresponding to shallow or steep climbs and dives and "level" flight. Finally, with four altitude "blocks" for the attacker, we could accommodate a 10,000 (20,000) ft. altitude variation in steps of 2500 (5000) ft.*

Reasonable control variables for this problem are load factor (n), bank angle (ϕ) and thrust level T. A fairly complete set of maneuvers could be obtained from $\{n=n_{max}, \phi=0,\pi,\pm\pi/2,\pm\pi/4; n=n_{max}/2, \phi=0,\pi,\pm\pi/2,\pm\pi/4; n=1,\phi=0\}$. If any of these maneuvers can be performed at $T=T_{max}$ or $T=0$,** we have 26 possible control choices per player (i.e., U=V=26).

The number of points/block required to calculate the transition probabilities with reasonable accuracy depends on the number of possible transitions. For the 3-D problem there are 19 possible non-zero p_{ij}'s for any state i.† In practice, we have found that for a given control pair it is common that only a few of the possible p_{ij}'s in a block are non-zero, a result that agrees well with physical intuition. Thus, P = 125 points/block should be adequate, especially if these points are selected randomly.

---

*The altitude of the target is simply $h_a + r \sin \eta$.

**Perhaps combined with speed brakes to give a deceleration.

†Because there are nine state variables and we assume no "corner" transitions.

Finally, for purposes of estimating computation time we shall assume that T=5, the value used in the PCP.

Using the above numbers, we obtain

$$C_{3D} \approx (.9 \times 10^6)(26)^2(125)(5)$$

$$\approx 3.75 \times 10^{11}$$

Similarly, using values appropriate to the VP gives

$$C_{VP} \approx (6.5 \times 10^4)(10 \times 10)(100)(5) \approx 3.25 \times 10^9$$

Extrapolation from Table 4 yields

$$(\text{CPU Time})_{3D} \approx \frac{3.75 \times 10^{11}}{8.17 \times 10^7} \cdot \frac{3.2 \times 10^4}{3.6 \times 10^3} \approx 4 \times 10^4 \text{ hrs.}$$

and

$$(\text{CPU Time})_{VP} \approx 3.5 \times 10^2 \text{ hrs.}$$

The above times are undoubtedly discouraging, but it must be remembered that we are talking about "global" solutions to highly complex games. Moreover, the estimates are for a relatively straightforward formulation of the problem and do not anticipate any improvements in the techniques employed. Let us now examine some reasonable assumptions concerning the problem that help reduce the computational requirements.*

*Later, we suggest some alternatives that retain many of the advantages of the Markov approach but are more limited in scope. These alternatives might provide quite useful results without imposing so severe a computational load.

A physically appealing assumption that we have not incorporated in the above formulation (except in the use of spherical coordinates) is to make the quality of information be range dependent. Thus, for example, we may assume that in the last two r-blocks (i.e., beyond 7500'), instantaneous relative heading cannot be resolved, implying the use of a single ψ-block for these ranges. This assumption alone reduces the number of state blocks, and hence the computation time, by 25%. Another highly tenable assumption is that the engagement terminates (or escape occurs) if the target is "behind" the pursuer (say, $135° < \xi < 225°$) and is headed in the opposite direction. This constraint reduces the number of state blocks by an additional 8%, so that 1/3 of the total computer time is saved by making both assumptions.

The original timing estimate for the 3D problem assumed that either player might use any one of 26 separate strategies in any block. It seems likely that careful analysis of the problem could eliminate many control choices that are undoubtedly nonoptimal in a given block. For example, if the target is off to the right, at long range and heading right, it is highly improbable that the attacker should turn left. While actual reductions in control choices will have to await a more detailed analysis of the problem, it seems reasonably conservative to assume that, <u>on the average</u>, one could eliminate 1/3 of a player's choices in each block.*
Such a reduction in control choices reduces the CPU time by more than a factor of 2.

---

*Necessary conditions might be used in the process. For example, given the nature of optimal solutions, it would not be surprising if the $\frac{n\ max}{2}$ controls did not satisfy necessary conditions. Eliminating these controls would reduce a player's possible choices from 26 to 14.

The net effect of the above assumptions is approximately an order-of-magnitude reduction in CPU time. Thus, solving the VP becomes reasonable, if expensive; the approximately 1.5 years CPU time makes it impractical to solve the 3D problem on a single CDC-6600. However, we again point out that we have assumed no improvements in the method, in coding efficiency, or in the computer itself. We already have some evidence of time-saving via coding efficiency. In particular, the use of the CDC "extended compiler" reduced computation time for the Homocidal Chauffeur problem by 20%. Because the same operations are repeated so many times in computing the transition probabilities, even a small improvement in coding efficiency yields a fairly substantial payoff.

The next generation computers will undoubtedly have a substantial speed advantage over the CDC-6600 and a 3D problem as described may well be within their capability. In this connection, it should be noted that the computation of the transition probabilities need not be done sequentially. Inasmuch as we are not core-limited, this problem could benefit greatly from the use of parallel processing.* Indeed, it would appear that a computer like the ILLIAC IV would be ideal for solving this type problem and would provide perhaps two orders of magnitude saving in CPU time.

*Note that this means that we could divide the burden of computing the transition probabilities among computers if more than one is available (as at LRC). This would reduce elapsed time to solve the problem.

7. CONCLUDING REMARKS

In this report we have presented the central results of a study to extend and further test the Markov game approach to aerial combat problems. The investigation involved development of a sophisticated computer program for solving Markov games pertaining to planar combat and application of that program to two idealized and one fairly realistic problem.

Numerical investigations of the idealized (Homocidal Chauffeur and Two-Car) games were directed at exploring fundamental aspects of the approach. We found that for the Homocidal Chauffeur, as the discretization became finer, the solution to the discrete Markov game approached the differential game solution in an important respect; namely, the discrete strategy barrier approached the continuous barrier. We also found that min-max and max-min solutions of the problem approached one another more closely as the discretization became finer. For the Two-Car problem, min-max and max-min results were very close even for a relatively crude discretization. These results increase our confidence in the basic approach.

Trajectories were also computed for the idealized problems and they revealed two interesting features of the solutions. First, we found that "chattering" between discrete state blocks with different optimal controls was a distinct possibility. Second, it was discovered that under certain conditions the evader could apparently escape, even though the expected time-to-capture was finite. Possible reasons for these behaviors were suggested.

The method was also applied to a highly nonlinear, five-state planar combat problem. It is fair to say that obtaining the feedback strategies for a problem of this magnitude represents a considerable advance in solving dynamic games; indeed, feedback solutions to optimal control problems of this size and complexity are rare. Thus, it is feasible to solve quite complex aerial combat games with this method. This capability for solving complex problems could prove most useful for systematic determination of significant aircraft design parameters such as wing loading, thrust loading, etc.; because both players use their optimal strategies, the effects of inadequate piloting techniques and of wide variations in those techniques do not obscure the basic issues involved. The Markov approach should also be helpful in evaluating tradeoffs between improved sensing capabilities (i.e., information) and other design parameters.

Suggestions for Further Research

Although the Markov approach in its present form may be used for fairly complex problems, extrapolation of our results has shown that it is presently impractical to solve a 3D-aerial combat game (of the scope described in Chapter 6) on present generation computers. Of course, such solutions could be made possible by developing new algorithms, particularly for computing transition probabilities. Parallel processing machines, such as ILLIAC IV could also advance the possibility. These avenues of approach should undoubtedly be explored. Rather than do that here, we suggest some alternatives that seem to have potential for more immediate payoff in studies of tactical maneuvering of aircraft. These alternatives retain the more important features of the Markov formulation in that they include the effects of imperfect information directly and they yield feedback solutions. The different approaches have in common an attempt to formulate and solve meaningful, but more tractable, sub-problems of the "global 3-D game." In a sense, they may be viewed as a means of avoiding the computation requirements imposed by having to consider all possible combinations of discrete states (N) and control pairs (UxV).

The M-Stage Game — A Transient Problem: In the game problems we have considered up to now, we have sought the optimal steady-state control policies, i.e., we have solved an ∞-stage game. The conceptual basis of an M-stage game is to determine the optimal pursuer-evader strategies for those states that have a nonzero probability of capture in M stages (or within a given fixed-time interval). As a result, it is not necessary to consider all states in the space, but only those that can be linked, through a sequence of M control decisions, to a capture state. There are several advantages to solving an M-stage game, as opposed to the steady-state problem. The nature of the computation is such that one constructs a Markov chain through the state space with capture blocks as absorbing states. Because the computation proceeds sequentially (starting at step 1), state transition probabilities are computed as they are needed and not a priori. Thus, if at a given stage a state has zero probability of capture, it is not necessary to compute its associated transition probabilities. On the other hand, as $M \to \infty$ there is no computational saving since most all states will have a nonzero probability of capture in the steady-state.

In the M-stage problem the cost functional assumes a different form. Expected time to capture is replaced by expected time to capture plus a term proportional to the probability of escape.*

*Escape is defined as being in any state other than capture after M stages.

Thus,

$$V = V_{max} \cdot (\text{Pr. escape}) + T_c$$

This is a meaningful criterion. The pursuer has the option of trading time to capture for more certain capture. Note also that as $M \to \infty$ in the steady-state the Pr. escape $\downarrow 0$ (the evader can be caught in the end) and the cost functional reduces to the expected time to capture.

There are various modifications to the basic M-stage problem that allow for even greater savings in computational effort. For example one may wish to delete low probability linkages. At any iteration there will be states that link (under optimal play) in only a weak manner (low transition probability) to favorable, low cost, states. It then becomes desirable to set these small transition probabilities to zero. The result is that one need consider fewer states at the next iteration. More important however is that the states that are considered (because of the deletion of low probability links) form a Markov chain that has strong bonds to capture blocks, i.e., a strong Markov chain. Thus we optimize over the most likely state transitions. This, along with the transient nature of the approach, should help to avoid the non-capture, closed trajectories we found in this study.

Local Solutions or Optimization in a "Tube": In formulating the Markov game we consider the potential range of variation of each state variable and then discretize according to physical considerations. If N_i is the number of discrete blocks associated with the i-th state variable, then the "playing" space is given by

$$N = \prod_{i=1}^{n} N_i$$

where n = no. of state variables (9 in the 3D-problem defined above). We then find the "global" optimal feedback solution, i.e., the control to choose in each of the N-blocks. This is a powerful solution, but it is often more than is actually needed. All the state blocks implied by the above combinatorial equation are not of equal practical interest.*

A systematic, practical and interesting way in which the state-space may be reduced is by restricting consideration to a region or "tube" (in n-dimensional space) surrounding a "nominal" trajectory. We then seek, in essence, the best feedback strategies in the "neighborhood" of that nominal trajectory. Our computational approach is well-suited to this task. It will be recalled that we treat a "compactified" region of the state-space as the "playing space." In this case, we simply let the compactified region be the "tube" of interest. An especially useful feature is that we can expand the tube incrementally, if such is indicated, without having to recompute transition probabilities (because "blocks" are processed independently in computing transition probabilities).

This approach should result in a much more tractable computational problem. While precise estimates are difficult to make without specifying particular "nominal" trajectories, our previous studies indicate that "optimal" trajectories in the planar problems traverse only a small fraction of the total number of state blocks. Moreover, one would expect that limiting the region of interest would reduce the number of control alternatives as well. Thus, it seems reasonable that "local" solutions to 3D-problems might be obtained with computer resources that are comparable to those used in obtaining the "global" solution to the planar combat problem.

*We have already seen how some states might be eliminated because they correspond to termination of the engagement through "escape" (and we were conservative in that elimination).

Strategy Optimization and Evaluation: We recall that in our present formulation the computational burden depends, in a combinatorial way, on the number of control choices for each player. If U and V are the number of choices for pursuer and evader, respectively, then transition probabilities for any state-block must be computed for UxV possible pairs. In addition, since a feedback strategy corresponds to a control choice for each state-block, there are U^N and V^N possible "strategies" for the pursuer and evader, respectively. While the optimization process fortunately does not rely on direct evaluation, the large number of potential strategies makes determination of the optimal more difficult and costly. Thus, any methods that can reduce the number of control choices for either or both players will have a substantial impact on the computational requirements.

Several possibilities for reducing the control choices suggest themselves. For example, the use of necessary conditions for optimality may eliminate some alternatives. Stipulating additional control constraints as a function of the state-blocks, either through physical or heuristic arguments*, is another strong possibility. Here we discuss two direct methods for reducing the computational load associated with too many "control pairs".

1. Strategy Evaluation and "One-Player" Optimization

The number of control pairs to be evaluated is drastically reduced if we fix the feedback strategy of one player.** Our computation procedure will then yield the opposing optimal feedback

*Such as those advanced in discussing the 3D-problem.

**We could also specify an open-loop control, or what is essentially the same, a trajectory for a player. Such might be appropriate for the target but doesn't appear to be for the attacker.

strategy. This, of course, removes some of the gaming aspects of the problem. However, it does allow one to evaluate a given strategy or "tactical doctrine"* against optimal opposition or, conversely, to determine the best course of action against given doctrine.

2. Strategy Optimization

Currently, we iterate to determine which of the U (and V) control choices is optimal in each of the N-state blocks. As noted above, this amounts to choosing the optimal strategies from among U^N and V^N possibilities. An alternate approach is to pre-specify several possible feedback strategies for each player and then determine the min-max strategies from these. In essence, this imposes a constraint on the problem. Undoubtedly, the computation associated with the optimization portion of the problem would be reduced if the number of possible strategies is kept within reasonable bounds. What of the computation of transition probabilities? Theoretically, if the number of pre-specified strategies (say S) was equal to or greater than the number of possible control choices (i.e., U or V), one might not save at all in computing transition probabilities. In practice, however, even if S>U (or S>V), we would expect great savings. The reason for this is that many of the candidate feedback strategies would undoubtedly have large regions of the state space in which the control choices were identical; when such is the case in those regions, we need only compute the transition probabilities appropriate to the common control choices.

*Such as energy maneuverability strategies.

8. REFERENCES

1. Baron, S., Chu, K.C., Ho, Y.C., Kleinman, D.L., "A New Approach to Aerial Combat Games," NASA CR-1626, October 1970.

2. Kushner, H. and Kleinman, A., "Numerical Methods for the Solution of the Degenerate Nonlinear Elliptic Equations Arising in Optimal Stochastic Control Theory," IEEE, Trans. Autom. Control, Vol. AC-13, 1968.

3. Chamberlain, S.G., "Stochastic Differential and Discrete Games: Existence of Solutions and Computational Procedures," Ph.D. Dissertation, Brown University, Rhode Island, June 1969.

4. Kushner, H. and Kleinman, A., "Accelerated Procedures for the Solution of Discrete Markov Control Problems," IEEE, Trans. Autom. Control, Vol. AC-16, April 1971.

5. Howard, R., "Dynamic Programming and Markov Processes," Massachusetts Institute of Technology Press, M.I.T., Cambridge, Massachusetts, 1960.

6. Isaacs, R., Differential Games, John Wiley and Sons, Inc., New York, 1965.